

Leçon 4 : Images, graphiques et packages — Énoncé

Objectif

Créer une fiche d'exercices avec graphiques, questions numérotées, images et un package trouvé dans Typst Universe.

Dans typst.app, ouvrez 04-starter.typ, complétez le code directement dans ce fichier, puis comparez avec 04-target.pdf.

Le travail se fait dans le fichier starter : complétez les zones TODO et les environnements `#exo(...)` indiqués.

Rappels de syntaxe

simple-plot

```
#import "@preview/simple-plot:0.9.0": plot

#plot(
  xmin: -5, xmax: 5.2, ymin: -2, ymax: 10,
  width: 8, height: 6,
  (fn: x => x*x, stroke: blue + 1.5pt, label: $f$),
)
```

taskize — Questions numérotées

```
#import "@preview/taskize:0.2.5": tasks

// Basique : 1 colonne, numéros 1. 2. 3.
#tasks()[
  + Question 1
  + Question 2
]

// 2 colonnes
#tasks(columns: 2)[...]

// Lettres a) b) c)
#tasks(label-format: "a")[...]

// Espacement personnalisé
#tasks(row-gutter: 1.5em, columns: 2)[...]
```

Paramètres :

- `columns` : nombre de colonnes (1, 2, 3...)
- `label-format` : format des puces ("1.", "a)", "A.", "i.")
- `row-gutter` : espacement vertical entre questions
- `column-gutter` : espacement horizontal entre colonnes

zebraw

```
#import "@preview/zebraw:0.6.3": *
#show: zebraw
```

Lire une page Typst Universe

Quand vous découvrez un package, vérifiez :

- le nom exact et la version ;
- la date de mise à jour ;
- la version minimale de Typst ;
- l'exemple minimal ;

- les paramètres dont vous avez vraiment besoin.

Images et figures

Important : Les graphiques et images doivent être centrés avec `#align(center)[...]` :

```
#align(center)[
  #plot(...) // ou #image(...)
]

#figure(
  image("../medias/figure-exemple.svg", width: 50%),
  caption: [Légende],
) <fig-label>
```

Voir [@fig-label](#).

Exercices

Lecture graphique

Exercice 1 Dans l'environnement `solution` du starter, tracer la parabole $f(x) = x^2 - 4x + 3$ avec `simple-plot`, puis créer 4 questions avec `taskize` sur 2 colonnes.

Système d'équations

Exercice 2 Dans l'environnement `solution`, tracer les deux droites du système suivant avec `simple-plot`, en utilisant deux couleurs différentes :

$$\begin{cases} y = 2x - 1 \\ y = -x + 4 \end{cases}$$

QCM avec taskize

Exercice 3 Dans l'environnement `solution`, créer un QCM avec `taskize` pour la question « Quelle est la valeur de $f(0)$? » : 4 propositions sur 2 colonnes, avec des lettres a) b) c) d) comme puces.

Calcul mental

Exercice 4 Dans l'environnement `solution`, créer une liste de 6 calculs sur 3 colonnes avec un espacement de 1.5em entre les lignes.

Algorithme Python

Exercice 5 Afficher du code Python avec coloration syntaxique via `zebraw`.

Figure avec image

Exercice 6 Insérer l'image fournie `../medias/figure-exemple.svg` avec légende, label et référence. Quand cette version compile, vous pourrez refaire la même chose avec une image personnelle uploadée dans `medias/`.

Découverte de package : QR code

Exercice 7 Chercher un package de QR code dans Typst Universe, lire sa documentation, puis ajouter un QR code vers `https://typst.app` dans votre document.

Dans votre rendu, ajoutez aussi une phrase courte expliquant pourquoi vous avez choisi ce package.

QR code cliquable

Exercice 8 Rendre le QR code de l'exercice précédent cliquable : un clic dessus doit ouvrir <https://typst.app>. Indice : la fonction `#link(...)[...]` rend cliquable tout son contenu, y compris une image. Aller voir ici pour plus d'information : <https://typst.app/docs/reference/model/link/>.

Checklist

- ☐ Graphiques avec bornes adaptées
- ☐ Courbes avec couleurs distinctes et labels
- ☐ Questions avec `taskize` (colonnes, format puces)
- ☐ Code coloré avec `zabraw`
- ☐ Image avec légende et référence
- ☐ Package de QR code trouvé via Typst Universe
- ☐ QR code ajouté et documenté par une phrase
- ☐ QR code rendu cliquable avec `#link(...)`