

Table des matières

1	Introduction	3			
2	Installation de \LaTeX	4			
2.1	Installer une distribution	4			
2.2	Ajouter un localtexmf	4			
2.3	Installer VSCode	4			
2.4	Installer les extensions VSCode	4			
	Exemple(s) - Organiser vscode	6			
	Exercice 1 - Le fameux 'Hello world!'	8			
3	Outils de formatage du texte	9			
3.1	Commandes usuelles	9			
	Méthode - Formatage du texte	9			
	Exercice 2 - Formatage du texte	9			
3.2	Commandes personnelles	9			
	Méthode - Définir une commande	9			
	Exercice 3 - Définir une commande	10			
3.3	Environnements usuels	10			
	Méthode - Formatage avec environnements	10			
	Exercice 4 - Mise en page avec environnements	11			
3.4	La géométrie	11			
	Méthode - Géométrie et \LaTeX	11			
3.5	Le mode mathématiques	11			
	Méthode - Mode mathématique	11			
4	Utiliser un package didactique	12			
4.1	Introduction	12			
4.2	Se documenter	12			
	Exercice 5 - Utiliser la documentation	12			
4.3	Compiler un document avec bfcours	12			
	Méthode - Première compilation avec bfcours	12			
	Exercice 6 - Première compilation avec bfcours	13			
4.4	Utiliser les environnements didactiques	13			
	Exercice 7 - Utiliser les environnements de bfcours	13			
	Aide - Utilitaire	13			
	Méthode - Zones de réponse	14			
				Exercice 8 - Environnements de bfcours	14
			4.5	Mise en page avancée	15
				Méthode - Grilles et structures	15
				Exercice 9 - Structure en grille	15
				Exemple(s) - Environnement MultiColonnes	16
5	Ateliers	17			
				Activité - Élaborer un document \LaTeX en groupe	17
				Exercice 10 - Utiliser Mathalea	17
				Exercice 11 - Construire une séquence	18
				Exercice 12 - Construire une évaluation	19
6	Atelier avancé : l'IA pour LaTeX	20			
6.1	Mise en place	20			
	Exercice 13 - Utiliser l'IA pour LaTeX	20			
6.2	Intégration de l'IA multi niveaux	20			
	Méthode - Les grands principes	21			
	Aide - Mettre en place Claude	21			
	Aide - Utiliser Windsurf ou Cursor	21			
	Aide - Github Copilot	21			
6.3	Guidelines pour prompts structurés (2025)	22			
	Définition - Prompts structurés modernes	22			
	Méthode - Structure XML + Markdown	22			
	Méthode - Ordre optimal des sections	22			
	Exemple(s) - Prompt complet structuré	22			
	Méthode - Techniques avancées	23			
	Méthode - Prompt optimisé	23			
6.4	Adapter bfcours à ses besoins	24			
	Méthode - Créer votre propre package	24			
	Méthode - Personnaliser un package	24			
6.5	Logiciel comme générateur de code LaTeX	24			
7	Annexes	25			
7.1	Exemple d'utilisation de Claude	25			
8	Correction des exercices	26			
	Définition - Fraction	28			

Nombre d'heures	6 heures			
Contenu de la séquence	MI	MF	MS	TBM
• Se familiariser avec les environnements de travail \LaTeX				
• Comprendre le fonctionnement général d'un document \LaTeX				
• Utiliser l'environnement EXO				
• Utiliser les exercices générés par la plateforme MathAléa				
• Utiliser les environnements didactiques de BFcours				
• Utiliser des outils numériques pour faciliter l'utilisation de \LaTeX				
• Mettre en place un agent IA dédié à \LaTeX				
• Créer son propre prompt optimisé				
• Créer son propre package				

Vocabulaire utilisé

• MikTeX (p. 3)	• se documenter (p. 3)	• IDE (p. 12)
• VSCode (p. 3)	• BFcours (p. 3)	• LLM (p. 20)
• LaTeX Workshop (p. 3)	• MathAléa (p. 3)	• prompts structurés (p. 21)
• PDF Viewer (p. 3)	• Python (p. 3)	• structure logique (p. 22)
• Commandes (p. 3)	• Live Share (p. 4)	• formatage du contenu (p. 22)
• Environnements (p. 3)	• vocabulaire (p. 9)	• Balises sémantiques (p. 22)
• CTAN (p. 3)	• mode mathématique (p. 11)	
• LaTeX Stack Exchange (p. 3)	• compiler (p. 12)	

1. Introduction

Introduction



- Plan.
- Comment fonctionne \LaTeX .
- Téléchargement des logiciels **MikTeX** et **VSCode**.

À faire :


- Se connecter à un point d'accès mobile.
- Télécharger les ressources de la formation.

Activité



- Setup des logiciels.
- Point théorique sur la structure d'un document \LaTeX .
- Le fameux « Hello World! »

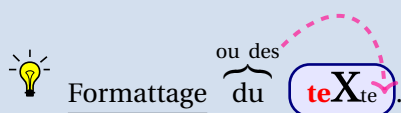
À faire :

- Télécharger l'extension **LaTeX Workshop** et **PDF Viewer** sur VSCode.
-  Télécharger le fichier « setup_vscode.json » pour configurer LaTeX Workshop.
- Construire son premier document \LaTeX .

Point théorique





- **Commandes et Environnements**



- **CTAN** et **LaTeX Stack Exchange**

À faire :

-  Se familiariser avec les commandes basiques.
-  Savoir où **se documenter**.

BFcours



- Une pause s'impose!
- Point théorique sur le package **BFcours**.
- Premier document avec **BFcours**.

À faire :

- Téléchargement du package **BFcours**.
- Compiler un premier document avec le package **BFcours**.

Ateliers



- Utiliser **MathAléa**.
- Utiliser les outils **BFcours**.
- Construire une séance avec **BFcours**.
- Construire une évaluation avec **BFcours**.

À faire :

- Construire une fiche d'exercices.
- Construire une séance de cours.
- Utiliser les outils **BFcours**.

Ateliers avancés



- Adapter **BFcours** (et \LaTeX en général) à ses besoins.
- Utiliser le générateur de questions Flash de **BFcours**.
- Utiliser Python pour générer des questions aléatoires.

À faire :

- Utiliser les fonctionnalités avancées de \LaTeX .
- Explorer le potentiel des combinaisons de **Python** et de \LaTeX .

2. Installation de \LaTeX

2.1 Installer une distribution

1. Aller sur la page de téléchargement de **MikTeX** - <https://miktex.org/download> et choisir la version **adaptée à votre système d'exploitation**.
2. **Cocher** l'option **installer les packages à la volée** (on-the-fly) pour permettre plus de souplesse dans les premières compilations.
3. **Décocher** l'option d'installation pour tous les utilisateurs. Cela rend plus simple l'utilisation de la console MikTeX.

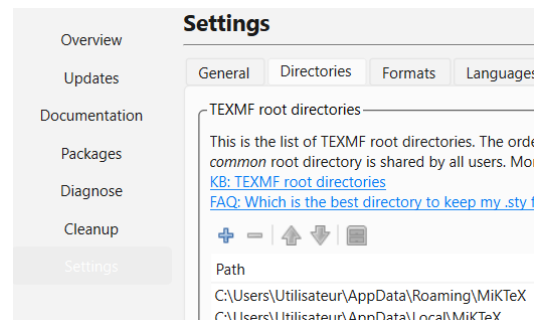
2.2 Ajouter un localtexmf

Il s'agit d'un **répertoire** respectant une **structure précise** qui, une fois configuré est automatiquement utilisable par le compilateur \LaTeX comme **dossier de packages**.

Suivre les étapes suivantes **une seule fois** :

```
localtexmf/  
├── tex/  
│   └── latex/  
│       └── MonPackage/  
│           ├── MonPackage.sty ..... fichier principal  
│           └── fichier_de_package.sty
```

1. Coller le dossier **localtexmf** récupéré sur ma **page GitHub** **n'importe où sur votre machine**. L'essentiel est qu'il reste à cet emplacement.
2. Copier le chemin d'accès de ce dossier.
3. Ouvrir la **console MikTeX** et aller au menu **Settings**.
4. Aller dans l'onglet « Directories ».
5. Appuyer sur le bouton **+** et **coller** le chemin d'accès au dossier **localtexmf**.
6. Confirmer les changements et quitter la console.



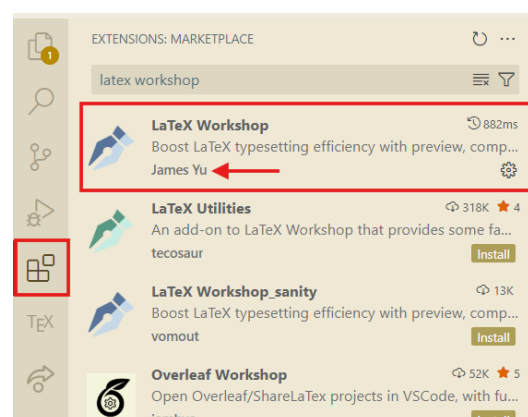
2.3 Installer VSCode

Le logiciel VSCode peut être remplacé par un autre IDE s'appuyant sur cette technologie comme **windsurf** ou **cursor**.

1. Aller sur la page de téléchargement de **VSCode** - <https://code.visualstudio.com/download> et choisir la version **adaptée à votre système d'exploitation**.
2. Laisser dans un premier temps les paramètres par défaut.
3. Il est possible de consulter des **tutoriels** en vidéo pour éditer le **style** de l'IDE. Tout est personnalisable.

2.4 Installer les extensions VSCode

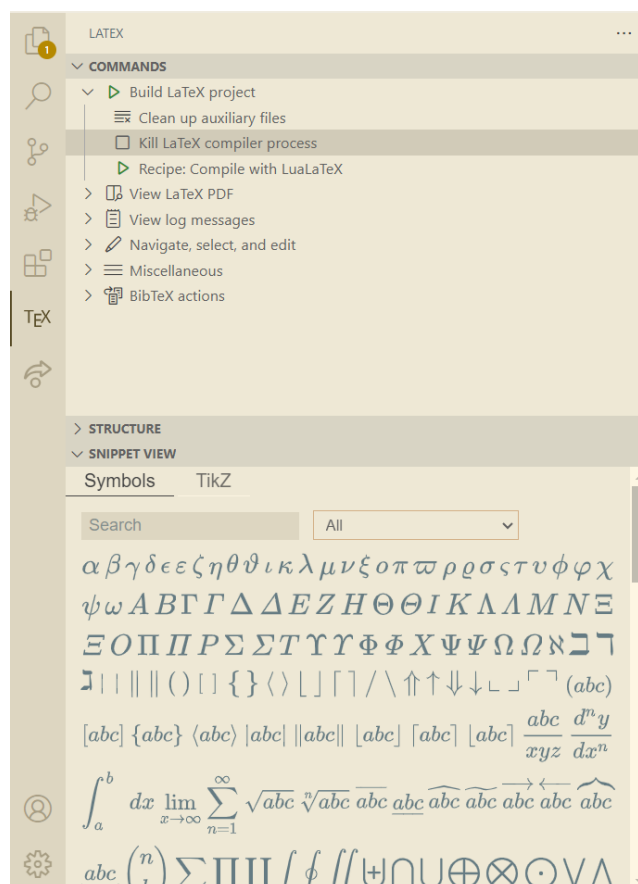
Puisque VSCode est un outil à **destination des développeurs**, il dispose de nombreuses extensions. Il convient d'explorer la bibliothèque d'extension qui prend la forme d'une **marketplace**. Par **précaution**, on se limitera à des extensions téléchargées de nombreuses fois et ayant **plusieurs étoiles**.



Il y a assez peu d'extensions à télécharger :

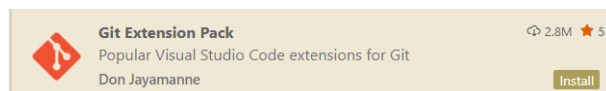
1. LaTeX Workshop

Inclut toutes les fonctionnalités de compilation automatique, visualisation de pdf.



2. git

Pour le suivi et la gestion des modifications. C'est un incontournable surtout lorsqu'on utilise activement des agents IA.



Remarque(s) :

Il est nécessaire d'**expliquer** à VSCode vos préférences de compilation.
Pour cela :

a. Ouvrir les **settings** de l'extension **LaTeX Workshop**.

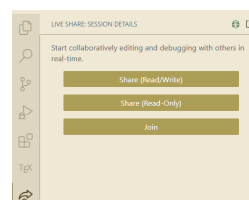
Barre de recherche \Rightarrow `>latex workshop settings`
 \Rightarrow `>settings Sync : Open User Settings(JSON)`

b. Ouvrir le fichier json de settings et **coller** le contenu du fichier ci-après.

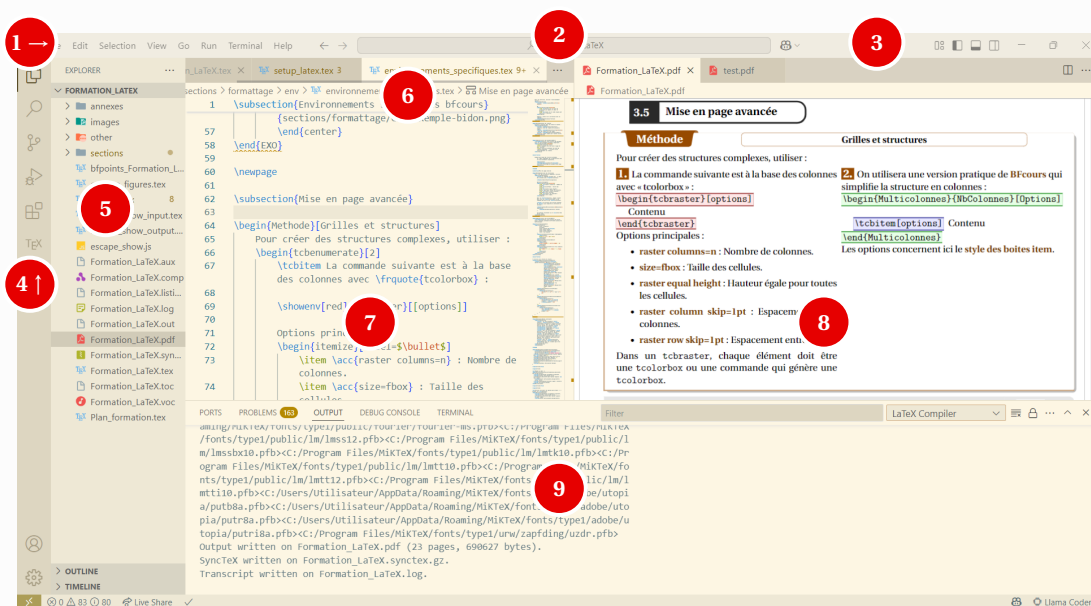
c. Désormais, à chaque sauvegarde, le fichier se compile automatiquement et son affichage dans le prévisualisateur pdf est automatiquement actualisé.

3. Sur VSCode, il est possible de **partager une session de travail** entre plusieurs participants via l'extension **Live Share**.

Il suffit de lire le « README » du projet pour se rendre compte de la facilité d'utilisation.



Une fois configuré, votre environnement de travail devrait ressembler à ceci :



On observe :

- | | |
|-------------------------------------------------------------------------------------|---------------------------------------------|
| 1. Options générales | 2. Barre de recherche |
| 3. Géométrie voulue pour le terminal | 4. La sidebar |
| 5. La zone d'exploration (fichiers, extensions, recherche sur fichiers multiples) | 6. Onglets |
| 7. Zone de saisie principale | 8. Zone d'affichage ou de saisie secondaire |
| 9. Terminal - permet également d'afficher les logs :
OUTPUT → Latex Compiler | |

```

{
  "terminal.explorerKind": "external",
  "terminal.external.windowsExec": "\"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\"",
  "terminal.integrated.shellIntegration.enabled": false,
  "terminal.integrated.env.windows": {
    "PATH": "${env:PATH};C:\\Program Files (x86)\\sox-14-4-2"
  },
  "latex-workshop.latex.tools": [
    {
      "name": "lualatex",
      "command": "lualatex",
      "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "%DOC%"
      ]
    }
  ],
  "latex-workshop.latex.recipes": [
    {
      "name": "Compile with LuaLaTeX",
      "tools": ["lualatex"]
    }
  ],
  "latex-workshop.latex.autoBuild.run": "onSave",
  "editor.wordWrap": "on",
  "editor.cursorBlinking": "expand",
  "editor.cursorSmoothCaretAnimation": "on",
  "workbench.iconTheme": "material-icon-theme",
  "editor.mouseWheelZoom": true,
  "github.copilot.enable": {
    "*": false,
    "plaintext": false,
    "markdown": false,
    "scminput": false,
    "latex": false
  },
}

```

1. Ouvrir le fichier `fichiers_de_la_formation/1.Exercices_formattage/hello_world/hello_world.tex`.
2. Appuyer sur `CTRL+s` ou `Compiler` pour compiler ce document.

Structure d'un document latex

```
\documentclass[a4paper,11pt,fleqn]{article} % Définition de la classe du document
```

% Définition de la géométrie du document

```
\usepackage[left=1cm,right=1cm,top=0.5cm,bottom=2cm]{geometry}
```

% Appel des packages

```
\usepackage[french]{babel} % Options spécifiques à la langue française
```

```
\usepackage{xcolor} % Utilisation des couleurs
```

```
\usepackage{fourier} % Police de caractère
```

% Définition de commandes

```
\newcommand{\tester}{test}
```

% Début du document

```
\begin{document}
```

Hello world!

```
\tester % Écrira 'test'
```

```
\end{document} % Fin du document
```


3. Outils de formatage du texte

3.1 Commandes usuelles

Méthode

Formatage du texte

Les commandes suivantes permettent d'effectuer la plupart des opérations sur le texte.

- | | |
|-------------------------------------------------------------|-----------------------------------------------------------|
| 1. <code>\underline{texte}</code> Souligner. | 2. <code>\hl{texte}</code> Surligner. |
| 3. <code>\textbf{texte}</code> Mettre en gras. | 4. <code>{\color{couleur}texte}</code> Mettre en couleur. |
| 5. <code>\acc[couleur]{texte}</code> Accentuer (bf). | 6. <code>\voc[couleur]{texte}</code> Vocabulaire (bf). |
| 7. <code>\textsc{texte}</code> Vocabulaire. | 8. <code>\fbox{texte}</code> Encadrer. |
| 9. <code>\emph{texte}</code> Mettre en italique. | 10. <code>\frquote{texte}</code> Citer. |
| 11. <code>\surligner[couleur]{texte}</code> Surligner (bf). | 12. <code>\encadrer[couleur]{texte}</code> Encadrer (bf). |

■ Exercice 2 – Formatage du texte



/ 2

1. Ouvrir le fichier

fichiers_de_la_formation/1.Exercices_formatage/premier_document/premier_document.tex

2. Reproduire la phrase suivante dans laquelle **chaque commande** est utilisée (une seule fois) :

En mathématiques, on peut souligner les éléments importants, **mettre en gras** ou **accentuer** des mots-clés, *mettre en italique* les théorèmes, **surligner** - ou bien **surligner** - des résultats, utiliser les PETITES CAPITALES ou la **commande** « voc » pour le **vocabulaire**, et **encadrer**, ou encore **encadrer** les formules essentielles.

3.2 Commandes personnelles

Méthode

Définir une commande

Pour définir une macro on peut utiliser la syntaxe ci-dessous.

1. `\newcommand{<N>}[<Nb>][<V>]{<C>}`

Où :

- **<N>** est le **nom** de la commande précédé d'un « backslash ».
- **<Nb>** est le nombre de paramètres
- **<V>** est la **valeur par défaut** du premier paramètre.
- **<C>** est le contenu de la commande.

Dans ce cas le premier paramètre peut être optionnel et assigné à une valeur par défaut.

Utiliser pour les commandes simples.

2. `\NewDocumentCommand{<N>}{<P>}{<C>}`

Où :

- **<N>** est le **nom** de la commande précédé d'un « backslash ».
- **<P>** sont les paramètres définis par O{valeur-ParDéfaut} pour les paramètres optionnels, et **m** pour les paramètres obligatoires.
- **<C>** est le contenu de la commande.

Utiliser pour les commandes complexes.

1. Définir une commande sans paramètre permettant de :

- Afficher le texte « Unité non présente ».
- Le texte doit être en **gras**.
- Le texte doit être coloré en **rouge**.

2. Définir une commande a un paramètre permettant de

- Afficher le texte « Bonjour <p> » dans lequel <p> est le paramètre de la commande.
- Le texte doit être en **gras**.
- Le texte doit être coloré en **vert**.

3.3 Environnements usuels

Méthode

Formattage avec environnements

Les environnements suivants permettent d'effectuer la plupart des opérations de mise en page.

1. `\begin{center}`

Contenu

`\end{center}` Centrer un texte/contenu.

3. `\begin{flushright}`

Contenu

`\end{flushright}` Aligner à droite.

5. `\begin{tcolorbox}[options]`

Contenu

`\end{tcolorbox}` une boite.

7. `\begin{itemize}[label=\bullet]`

Contenu

`\end{itemize}` Listes à puces.

9. `\begin{tcbenumerate}[n][i]`

Contenu

`\end{tcbenumerate}` Listes numérotées sur n colonnes à partir de l'indice i de bfcours.

11. `\begin{align*}`

Contenu

`\end{align*}` Mode maths aligné (séparateur &).

13. `\begin{crep}`

Contenu

`\end{crep}` Cadre de réponse (bf).

2. `\begin{flushleft}`

Contenu

`\end{flushleft}` Aligner à gauche.

4. `\begin{multicols}{n}`

Contenu gauche

`\columnbreak`

Contenu droit

`\end{multicols}` Affichage sur n colonnes avec les packages standard.

6. `\begin{minipage}{0.475\textwidth}`

Contenu

`\end{minipage}` une petite page dans la page.

8. `\begin{enumerate}`

Contenu

`\end{enumerate}` Listes numérotées.

10. `\begin{tabular}[titre]{structure}`

Contenu

`\end{tabular}` Tableaux.

12. `\begin{tcctab}[titre]{structure}`

Contenu

`\end{tcctab}` Tableaux encadrés de bfcours.

14. `\begin{MultiColonnes}{n}[options]`

Contenu

`\end{MultiColonnes}` Disposition en n colonnes avec boites de style « options » (bf).

Reproduire la mise en page suivante en utilisant les environnements adéquats :

Les différents types d'alignements

Ce texte est aligné à gauche grâce à l'environnement `flushleft`.

Ce texte est aligné à droite grâce à l'environnement `flushright`.

Ce texte est centré grâce à l'environnement `center`.

Résumé des environnements d'alignement

Environnement	Description	Utilisation
<code>flushleft</code>	Aligne à gauche	Texte courant
<code>center</code>	Centre le texte	Titres, équations
<code>flushright</code>	Aligne à droite	Signature, date

3.4 La géométrie

Méthode

Géométrie et \LaTeX

L'utilisation de la géométrie repose essentiellement sur le package **TikZ**.

Son utilisation est **omniprésente** en \LaTeX - la bordure de cet environnement est **dessinée** avec une figure TikZ. Malheureusement, la maîtrise de ce package nécessiterait une formation à part entière (- cf. sa documentation CTAN et les ouvrages associés **TikZ pour l' impatient** ou bien le package **tkz-euclid**).

Néanmoins, le professeur de mathématiques sera ravi d'apprendre que le logiciel de géométrie dynamique **Geogebra** ou d'autres comme celui du groupe **coopmaths** permettent un **export TikZ** des figures réalisées.

Fichier → Exporter → Graphique vers PGF/TikZ

Il suffit ensuite de se laisser guider par l'interface proposée par Geogebra.

On veillera à copier coller le contenu généré **entre les bornes** `\begin{document}` et `\end{document}`.

L'utilisation d'un **script de reformattage** des figures TikZ ainsi générée est **hautement conseillé** - les outils de **BFcours** proposent un tel programme adapté à plusieurs situations.

3.5 Le mode mathématiques

Méthode

Mode mathématique

Le **mode mathématique** permet l'accès aux commandes de calcul et **adapte** la police aux mathématiques.

Pour une **documentation**, on peut conseiller : **Petit guide des mathématiques - CTAN**. Ressource en anglais, mais comportant de nombreux exemples.

On l'utilise de plusieurs manières ayant chacune leur spécificité.

Mode basique

S'utilise via : `$ contenu maths $`
 → S'insère dans le texte.
 → Simple à utiliser.

Mode étendu

S'utilise via : `\(contenu maths \)`
 → S'insère dans le texte.
 → Tailles plus importantes (fractions).

Mode display centré

S'utilise via : `\[contenu maths \]`
 → Saute une ligne et indente.
 → Tailles plus importantes (fractions).

Mode equation / align

`\begin{align*}`
`contenu & maths`
`\end{align*}`

4. Utiliser un package didactique

4.1 Introduction

L'on dit toujours que le plus important lorsque l'on programme un logiciel ou un document, ce sont les **contenants**. En effet, de bons contenants automatisent certaines fonctionnalités que l'auteur souhaite retrouver en tous temps. C'est précisément ce qui bloque de nombreux adeptes de \LaTeX .

De nombreux packages sont disponibles et proposent des fonctionnalités plus ou moins équivalentes à celles développées dans **BFcours**.

Dans la suite de cette formation, nous utiliserons **BFcours** par soucis d'homogénéité de la formation, mais il est tout à fait possible d'utiliser un autre package didactique à la place.

Cela permet d'introduire la sous-section la plus importante de cette formation : **la documentation**.

4.2 Se documenter

■ Exercice 5 – Utiliser la documentation



/ 2

Explorez l'univers de la communauté \LaTeX en découvrant deux packages didactiques comme alternatives à **BFcours**.

1. Utiliser le package `pas-cours`.
 - Accéder à la documentation de `pas-cours` sur **CTAN** - « `pas-cours` ».
 - Repérer les environnements didactiques.
2. Utiliser le package `profMaquette`.
 - Aller sur le site <https://ctan.org/>
 - Chercher la documentation du package « `profMaquette` » de **Christophe Poulain**.

4.3 Compiler un document avec `bfcours`

Méthode

Première compilation avec `bfcours`

L'architecture recommandée pour éditer des documents \LaTeX est la suivante :

```
NomPrincipal
├── images/ ..... Dossier pour toutes les images
├── sections/ ..... Optionnel pour les gros fichiers
├── annexes/ ..... Optionnel, documents auxiliaires
│   ├── scripts/
│   ├── csv/
│   └── todolists/
├── NomPrincipal.tex ..... Fichier principal à compiler
├── enonce.tex ..... Contenu principal, organise les sous-fichiers
└── enonce_figures.tex ..... Contient les figures TikZ du projet
```

Les répertoires en `bleu` sont présents en cas de nécessité seulement.

Cette organisation permet de :

- Séparer clairement le contenu de la structure : les en-têtes sont dans le **fichier principal** et le contenu est dans **enonce.tex**.
- Produire différentes versions pour un même contenu (A3, dys, corrections, élève...)
- Faciliter la maintenance et les modifications
- Réutiliser des éléments entre différents projets

Pour **compiler** un document \LaTeX avec `bfcours`, il suffit de :

1. Écrire le contenu dans un fichier nommé **enonce.tex**
2. Compiler le **fichier principal** en utilisant **LuaLaTeX** - sinon ça ne compile pas.

Remarque(s) :

LuaLaTeX est un compilateur permettant de compiler avec le moteur \LaTeX tout en autorisant l'utilisation de scripts écrits en langage Lua.

Ce langage est simple et fonctionne sur tous les hardware, mais n'est pas autorisé par défaut dans les compilateurs \LaTeX .

Accéder au répertoire `fichiers_de_la_formation/2.Exercices_bfcours`.

Ouvrir son fichier principal `fichiers_de_la_formation/1.Exercices_bfcours/Exercices_bfcours.tex` avec **VScode** (ou l'IDE de votre choix).

Compiler le fichier.

BF-1

4.4 Utiliser les environnements didactiques

Le package **BFcours** propose des **environnements didactiques** destinés à la transmission de connaissances. Ils s'utilisent de manière très simple mais agissent à plusieurs niveau et sont l'aboutissement de beaucoup de travail.

- Présentation claire isolant le contenu du reste du document.
- Insertion dans la table des matières.
- Gestion des couleurs permettant une cohérence visuelle.
- Gestion des marges et de la police.
- Pour les exercices : gestion des numéros, des corrections séparées, des points et de la difficulté.

1. On peut donc utiliser tous les environnements suivants librement, via la syntaxe :

```
\begin{NomEnvironnement}[titre][options]
Contenu
\end{NomEnvironnement}
```

- **NomEnvironnement** : Le nom de l'environnement commence toujours par une majuscule et sans accent (Exemple : Methode, Definition, Theoreme).
- **Titre** : La première option correspond au titre.
- **Options** : La seconde option est destinée à ajouter des options tcolorbox dans la définition de l'environnement.

2. Environnements didactiques

Methode	Pour présenter des méthodes de résolution
Definition	Pour introduire une nouvelle définition
Theoreme	Pour énoncer un théorème mathématique
Exemple	Pour illustrer par un exemple
EXO	Pour proposer un exercice
Remarque	Pour ajouter une remarque
Notation	Pour définir une notation
Demonstration	Pour ajouter une démonstration
Activite	Pour ajouter une activité
Aide	Pour ajouter une aide.

Remarque(s) :

L'environnement **EXO** a une syntaxe particulière :

```
\begin{EXO}{Titre}{Code compétence}
Contenu
\end{EXO}
```

Écrire un fichier contenant la **Definition** d'une **fraction** en utilisant bfcours.

Aide

Utilitaire

On peut utiliser les commandes suivantes :

1. `\dfrac{Num}{Den}` - Pour les **fractions**.

S'utilise en **mode mathématiques**.

2. `\red{texte}` - Raccourci pour écrire en rouge.

3. `\Si` - L'un des **connecteurs logiques** de **BF-cours**.

4. `\Alors` - L'un des **connecteurs logiques** de **BFcours**.

BF-3

Dans de trop nombreuses ressources, la gestion des **espace réponse** reste trop superficielle. Les environnements du package **rdcrep** associés à ceux de **BFcours** forment un ensemble permettant d'éditer des zones de réponse bienveillants.

1. `\repsim[largeur]{texte}` % Cadre simple pour réponse courte.

Date : 20/04/25 Note : 10 /20

3. `\begin{crep}[extra lines=n]`
Contenu
`\end{crep}` % Zone de réponse sur plusieurs lignes.

Ceci est une réponse

sur plusieurs lignes.

2. `\tcfillcrep{texte}` % Zone de réponse extensive sur une ligne. Ceci est un texte.

La même commande sur une ligne vide

4. `\setrdcrep {`
seyes, % / colback=white,
correction=true,
correction color=monrose,
correction font = \large\bfseries,
tcolorbox_options % options supplémentaires
} % Style de la correction

■ Exercice 8 – Environnements de bfcours



/ 2

Créer une structure de document avec les environnements adéquats comportant :

- Une définition titrée
- Un exemple avec une mise en page en deux colonnes
- Un exercice contenant une zone de réponse permettant à l'élève d'écrire confortablement sa réponse.

Méthode

Grilles et structures

Pour créer des structures complexes, utiliser :

1. La commande suivante est à la base des colonnes avec « tcolorbox » :

```
\begin{tcbrafter}[options]
```

Contenu

```
\end{tcbrafter}
```

Options principales :

- **raster columns=n** : Nombre de colonnes.
- **size=fbox** : Taille des cellules.
- **raster equal height** : Hauteur égale pour toutes les cellules.
- **raster column skip=1pt** : Espacement entre colonnes.
- **raster row skip=1pt** : Espacement entre lignes.

Dans un tcbrafter, chaque élément doit être une tcolorbox ou une commande qui génère une tcolorbox.

2. On utilisera une version pratique de **BFcours** qui simplifie la structure en colonnes :

```
\begin{Multicolonnes}{NbColonnes}[Options]
```

```
\tcbitem[options] Contenu
```

```
\end{Multicolonnes}
```

Commentaires :

- Dispose le contenu en **grille** en le stockant dans des **tcolorbox**.
- La commande `\tcbitem` permet de changer de boîte - et donc de colonne.
- Les options de MultiColonnes modifient de façon globale le style des boîtes.
- Les options de tcbitem modifie de façon locale le style de la boîte.

Remarque(s) :

- Les options ci-dessus sont celles utilisables dans le package **tcolorbox**.
- Puisque **tout est boîte**, la documentation de ce package est **l'outil par excellence** pour produire du contenu de qualité et utiliser correctement les nombreuses options disponibles.

Exercice 9 – Structure en grille



/ 2

Créer une structure en grille avec 3 colonnes contenant différentes propriétés mathématiques :

ENV-3

Produit remarquable

$$(a+b)^2 = a^2 + 2ab + b^2$$

Identité trigonométrique

$$\sin^2(\alpha) + \cos^2(\alpha) = 1$$

Dérivée d'un produit

$$(u \times v)' = u' \times v + u \times v'$$

```
\begin{MultiColonnes}{2}[colframe=black,boxrule=0.4pt,halign=center]% Default [blank]
\tcbitem[valign=bottom] Hauteur adaptée pour chaque ligne.
\tcbitem[title=La seule boîte titrée,colframe=black,boxrule=0.4pt] Bonjour!
\tcbitem[raster multicolumn=2,halign=center] Fusion facile..
\tcbitem[colback=green!25!white] Ligne 3 colonne 1
\tcbitem Utilisation basique.
\end{MultiColonnes}
```

Rendu du code :

Alignement disponible.	La seule boîte titrée Hauteur adaptée par ligne .
Fusion facile..	
Ligne 3 colonne 1	Utilisation basique.

Version utilisant les options par défaut : (enlever les **options**)

Alignement disponible.	La seule boîte titrée Hauteur adaptée par ligne .
Fusion facile..	
Ligne 3 colonne 1	Utilisation basique.

Style modifié : en utilisant un `\tcbset` précis.

```
\tcbset {
  ColonnesBaseStyle/.style={
    top=0pt,
    bottom=0pt,
    left=0pt,
    right=0pt,
    colback=blue!5!white,
    colframe=blue!75!black,
    before title=\dimcoloredsquare{white}{1.5} ,
    after title=\hfill \today },
  boxrule=0.4pt
}
```

Alignement disponible.	■ La seule boîte titrée 26 mai 2025 Hauteur adaptée par ligne .
Fusion facile..	
Ligne 3 colonne 1	Utilisation basique.

Activité

Élaborer un document \LaTeX en groupe

À travers les exercices suivants, vous apprenez à utiliser l'**écosystème** \LaTeX ainsi que les facilités de partage de code.

Durée :  30'

`\overlaychrono{30}`

1. **Répartir** les participants en groupes de 3 ou 4 - pour la diversité, pas de contrainte particulière - par affinité.
2. Chaque groupe **effectuera** l'un des exercices. Si l'un des exercices est terminé, il est possible d'en choisir un nouveau.
3. Les groupes doivent **s'organiser** pour **répartir les tâches** entre les participants. Un **responsable** sera chargé de collecter les parties du code des membres du groupe.

L'utilisation des outils d'**intelligence artificielle** sont bien entendus autorisés.


■ Exercice 10 – Utiliser Mathalea



/ 2

Construire une fiche d'exercice sur le thème des fractions pour le niveau 6^{ème}.

Pour cela :

1. Ouvrir le répertoire `fichiers_de_la_formation/Atelier_exercices_mathalea` dans VScode.
2. Aller sur <https://coopmaths.fr/alea/> et **élaborer** une série d'exercices.
3. **Copier le contenu seul** généré dans le fichier `fichiers_de_la_formation/Atelier_exercices_mathalea/enonce.tex`
4.  Le document généré n'est pas adapté à la présentation de **BFcours**.
Il faut le formater à l'aide du programme `fichiers_de_la_formation/programmes/mathalea_adapter.exe` (double cliquer)
Ensuite, il suffit de choisir le fichier `fichiers_de_la_formation/Atelier_exercices_mathalea/enonce.tex` dans lequel vous aurez collé le contenu généré par MathAlea.
5. Le programme produit un fichier `fichiers_de_la_formation/Atelier_exercices_mathalea/enonce_TOOLS.tex` .
Il convient de **modifier** la ligne : `\input{enonce}` en `\input{enonce_TOOLS}` dans le fichier principal.
6. **Explorer** les possibilités offertes par \LaTeX en modifiant certains exercices et / ou leurs corrections de sorte à **utiliser** les **environnements de réponse** de **BFcours**.

A-1

Construire une séquence de cours sur le thème des **fractions** pour le niveau 6^{ème}.

Objectifs :

- Une définition
- Un exemple de fraction avec une figure (image ou tikz).
- Une méthode pour placer une fraction sur un axe gradué.

Pour cela :

1. Ouvrir le répertoire `fichiers_de_la_formation/Atelier_bfcours` dans VScode.
2. **Elaborer** rapidement un plan de séquence. **Répartir les tâches** pour que chaque participant ait une tâche spécifique.
3. Chaque participant complète sa partie dans un fichier.
4. Le responsable récupère les différents fichiers à la fin du temps imparti et s'occupe de produire la synthèse. On utilisera la commande `\input{nom_du_fichier.tex}` pour inclure les fichiers récupérés.
5. Le responsable **distribue** le document complet à ses collègues.


Construire une évaluation sur le thème des **fractions** pour le niveau 6^{ème}.

Durée : 30 minutes

Compétences testées :

- Définition des fractions
- Fraction associée à une figure
- Fraction sur un axe gradué simple
- Fraction quotient

Pour cela :


1. Ouvrir le répertoire `fichiers_de_la_formation/Atelier_evaluation` dans VScode.
2. Aller sur <https://coopmaths.fr/alea/> et **élaborer** une série d'exercices.
3. Copier le **contenu seul** généré dans le fichier `fichiers_de_la_formation/Atelier_evaluation/enonce.tex`
4.  Le document généré n'est pas adapté à la présentation de **BFcours**.
Il faut le formater à l'aide du programme `fichiers_de_la_formation/programmes/mathalea_adapter.exe` (double cliquer)
Ensuite, il suffit de choisir le fichier `fichiers_de_la_formation/Atelier_evaluation/enonce.tex` dans lequel vous aurez collé le contenu généré par MathAlea.
5. Le programme produit un fichier `fichiers_de_la_formation/Atelier_evaluation/enonce_TOOLS.tex`.
Il convient de **modifier** la ligne : `\input{enonce}` en `\input{enonce_TOOLS}` dans le fichier principal.
6. **Explorer** les possibilités offertes par \LaTeX en modifiant certains exercices et / ou leurs corrections de sorte à **utiliser** les **environnements de réponse** de **BFcours**.

A-3

6.1 Mise en place

La section suivante suit une structure différente du reste de cette formation.

Il s'agit de **méthodes** permettant d'utiliser l'intelligence artificielle pour toute opération concernant des documents LaTeX.

 Bien que le monde de l'IA évolue très rapidement, si vous n'utilisez pas l'IA en général, ce qui suit constituera un point d'entrée **robuste**.

Les guides de bonne pratiques utilisés sont issus des instructions **Anthropic** afin d'obtenir le meilleur du potentiel des modèles de langage.

On se restreindra aux pratiques validées par la communauté et qui semblent nécessaires dans le contexte précis d'utilisation de \LaTeX .

■ Exercice 13 – Utiliser l'IA pour LaTeX



/ 2

Après lecture de cette section :

1. Choisir un modèle d'IA.
2. Brainstormer quelques idées d'agents IA adaptés à vos besoins pour des documents spécifiques.
3. Concevoir un prompt **respectant les bonnes pratiques** pour que l'IA réalise **l'un de ces documents**
4. Préparer quelques demandes tester l'efficacité du modèle.
5. Produire un court rapport (éventuellement documenté avec images) de vos expérimentations et de vos retours.

IA-1

6.2 Intégration de l'IA multi niveaux

L'intelligence artificielle actuellement utilisée repose sur les **LLM**.

Ces modèles produisent du texte. Cela tombe parfaitement bien puisque le langage LaTeX fonctionne exclusivement grâce à du texte.

Il y a donc plusieurs niveaux sur lesquels l'IA peut agir :

Debuguer Sur copie d'un contenu \LaTeX la plupart des outils IA actuels peuvent : <ul style="list-style-type: none"> • Le corriger • Déterminer l'en-tete correspondante • Le modifier en suivant des instructions basiques 	Produire du contenu Sur base d'un modèle \LaTeX et d'une série d'instructions précises, certains modèles peuvent produire du contenu \LaTeX de qualité.
Recopier du contenu Sur base d'un modèle \LaTeX et d'une série d'instructions précises, certains modèles peuvent reprendre un document pdf ou image pour produire du contenu \LaTeX .	Créer des packages L'IA peut fabriquer des commandes ou environnements qui répondent à vos besoins spécifiques et les intégrer dans vos documents ou dans vos packages
Agent formateur Sur base d'un extrait de documentation, l'IA peut résumer les principales fonctionnalités disponibles et vous fournir celles dont vous avez besoin en lien avec vos demandes. On peut faire produire des rapports (markdown ou \LaTeX) pour que l'IA nous forme à utiliser certaines technologies. C'est de cette manière dont j'ai appris \LaTeX .	Agent modérateur Certains modèles de langage avancés sont capable de jouer le rôle d'agent principal en actionnant divers outils. Cela dépasse largement le cadre de cette formation, mais on peut envisager un agent IA qui irait se documenter sur internet, produire quelques diapos avec canvas, produire un genially d'une activité et produire une fiche d'activité \LaTeX pour donner les consignes aux élèves dans laquelle figureraient les qrcode.



Les solutions pour intégrer l'IA dans sa démarche de travail nécessitent un abonnement dont les prix vont - actuellement - de 10€ par mois (Github Copilot) à 200€ par mois (ChatGPT max).

La plupart des abonnements « pro » coûtent environ 20€ par mois et correspondent aux besoins d'un enseignant qui se forme au \LaTeX . Il suffit de choisir un modèle et de l'utiliser avec cette formule.

Méthode

Les grands principes

L'intégration de l'IA repose sur les principes suivants :

Un modèle

Parfois on préfère un **modèle rapide**, parfois un **modèle de réflexion**, parfois un **modèle d'action** et parfois un **modèle multimodal** (images, voix...)

Un système d'intégration

On peut **copier/coller** le code produit par le modèle. On peut également permettre à l'IA de produire son code sur votre ordinateur via un **logiciel**.

Des prompts

C'est le plus important. Ce prompt doit être :

- **Cohérent** Permet une réponse sans bugs et sans oubli.
- **Fourni en exemples de qualité** Votre style sera copié.
- **Structuré - XML / markdown** L'IA intégrera mieux les informations et en plus grande quantité.

Aide

Mettre en place Claude

L'entreprise Anthropic et son modèle **Claude 4-opus** sont à l'origine du **protocole MCP : Model Context Protocol**. Cela permet aux agents d'utiliser **n'importe quel outil numérique**.

Entre autres :

- Votre système de fichiers
- Aller sur internet
- Utiliser n'importe quel outil externe qui propose des fonctionnalités publiques destinées aux agents IA (canvas par exemple...)

J'utilise au quotidien ce modèle d'IA via **l'application locale**.

Je propose mes **prompts** comme base officielle pour utiliser **bfcours** avec Claude. Cela lui permet de préparer les documents avec mes exigences de qualité.

Prompts de bfcours

```
fichiers_de_la_formation/
├── ressources/
│   ├── prompts_claude_ai/
│   ├── fichiers_exemples_a_donner/
│   └── claude_config_mcp.json
```

L'intervention humaine de modification est bien sûr toujours possible. L'agent codeur est également à même de modifier par la suite le code qu'il a créé.

Aide

Utiliser Windsurf ou Cursor

Ce sont des IDE comme VScode qui intègrent naturellement l'IA dans leur fonctionnement. Soutenus par de grandes entreprises une grande communauté de développeurs. On peut utiliser le même **prompt system** que celui donné dans l'aide précédente.

Aide

Github Copilot

Ce modèle s'intègre bien dans VSCode et permet de la génération de code \LaTeX avec des suggestions. Je ne l'utilise pas, mais c'est une solution plus discrète adaptée à ceux qui souhaitent rester maître du code écrit. L'IA intervient pour **suggérer** des modifications **en contexte**. Il s'agit d'une alternative **peu coûteuse**.

Définition

Prompts structurés modernes

Les **prompts structurés** représentent l'évolution des techniques de prompt engineering en 2025, combinant la précision du **XML** pour la structure logique avec la lisibilité du **Markdown** pour le contenu. Cette approche permet d'obtenir des réponses plus fiables et cohérentes des modèles de langage comme Claude d'Anthropic.

Méthode

Structure XML + Markdown

Principe fondamental

- **XML** pour la **structure logique**
- **Markdown** pour le **formatage du contenu**
- **Balises sémantiques** descriptives

Exemple de base

```
<instructions>
## Tâche principale
- Analyser le document
- Extraire les points clés
- **Formater** la sortie
</instructions>
```

Méthode

Ordre optimal des sections

L'ordre des sections dans un prompt structuré suit une progression logique du général au spécifique :

- | | |
|----------------------------------------------|----------------------------------------------------|
| 1. <system_role> – Qui est l'agent | 2. <context> – Situation et environnement |
| 3. <data> – Données de référence | 4. <rules> – Règles et contraintes |
| 5. <examples> – Exemples few-shot | 6. <instructions> – Actions à effectuer |
| 7. <output_format> – Format de sortie | 8. <critical_reminders> – Points essentiels |

Exemple(s)

Prompt complet structuré

Voici un exemple de prompt utilisant toutes les bonnes pratiques :

```
?xml version="1.0" encoding="UTF-8"?
<agent_prompt>
<system_role>
Expert en analyse de données financières
</system_role>
<context>
<company>Startup technologique</company>
<period>Q1 2025</period>
<objective>Analyse de performance</objective>
</context>
```

```
<instructions>
## Analyser les métriques suivantes
1. Croissance du chiffre d'affaires
2. Taux de rétention clients
3. Coûts d'acquisition
**Produire** un rapport structuré
</instructions>
<output_format>
- Executive summary
- Analyse détaillée par métrique
- Recommandations priorisées
</output_format>
</agent_prompt>
```

Remarque(s) :

✗ À éviter

- ✗ Instructions négatives excessives (« Ne fais pas... »)
- ✗ Balises XML invalides ou mal fermées
- ✗ Mélange contexte/instructions
- ✗ Exemples incohérents en format
- ✗ Instructions incohérentes

✓ Bonnes pratiques

- ✓ Instructions positives claires
- ✓ Structure XML valide et cohérente
- ✓ Séparation nette des sections
- ✓ Exemples uniformes
- ✓ Concision, précision et cohérence

Méthode

Techniques avancées

Chain of Thought

Technique permettant de décomposer le raisonnement :

```
<thinking_process>
1. D'abord, analyser...
2. Ensuite, vérifier...
3. Finalement, produire...
</thinking_process>
```

Scratchpad

Pour les documents longs, utiliser un bloc-notes :

```
<scratchpad>
Points clés à retenir :
- Information critique 1
- Donnée importante 2
</scratchpad>
```

Scratchpad permanent :

```
<scratchpad>
Utilise le fichier mon_repertoire_IA/AI_NOTES.xml pour :
- Noter ton plan d'action
- Tenir à jour les étapes effectuées au fur et à mesure.
</scratchpad>
```

Cette dernière méthode est extrêmement efficace et permet aux agents de se succéder de façon fluide.

Méthode

Prompt optimisé

Adapter la structure selon le type de tâche :

Pour du code

- Structure de fichiers attendue
- Conventions de nommage
- Exemples d'entrée/sortie

Pour des documents

- Templates de structure
- Styles d'écriture
- Ton et registre

Pour des analyses

- Critères d'évaluation
- Métriques à calculer
- Format de rapport

Pour un agent spécialisé en \LaTeX avec **bfcours**, voici la structure optimale :

```
<latex_agent_prompt>
├─ <system_role> ..... Expert LaTeX/bfcours
├─ <context> ..... Environnement pédagogique
├─ <bfcours_conventions>
│   ├── <environments> ..... Definition, Exemple, etc.
│   ├── <commands> ..... \voc, \acc, etc.
│   └── <layout> ..... MultiColonnes, tcraster
├─ <quality_requirements>
│   ├── <imperative_rules> ..... Règles strictes
│   ├── <design_principles> ..... Principes de design
│   └── <workflow_instructions> ..... Patterns de travail / chaîne de pensée
└─ <critical_reminders> ..... Points clés à respecter (synthétique)
```

Processus d'itération

1. **Tester** avec des cas limites variés
2. **Analyser** les échecs et réussites
3. **Enrichir progressivement** la structure et les instructions
4. **Affiner** le prompt
5. **Conserver** une bibliothèque de prompts testés
6. **Partager** les agents efficaces découverts

Règle d'or : Un prompt bien structuré = Moins de corrections + Meilleure productivité

« La structure parfaite émerge de l'expérimentation continue. »

Méthode

Créer votre propre package

Il convient de créer votre propre package pour apporter vos modifications de façon globale. (voir la section **Ajouter un localtexmf**)

Il suffira alors de l'utiliser dans vos documents :

```
\usepackage{bfcours}
```

```
\usepackage{adapt-bfcours} % Votre package qui modifiera bfcours
```

Méthode

Personnaliser un package

Puisque \LaTeX est orienté vers la personnalisation, il est possible d'adapter n'importe quel package à vos besoins. Il y a plusieurs façons de procéder :

1. Créer vos propres commandes qui simplifient l'utilisation de celles données dans les packages utilisés.

```
\newcommand{\bonjour}{bonjour} % définition
```

d'origine

```
\newcommand{\mybonjour}{Bonjour !} % définition que vous utiliserez
```

2. Réécrire certaines commandes pour qu'elles agissent différemment.

C'est un peu plus compliqué, il faut retrouver le code d'origine de la commande, recopier son contenu et modifier la copie.

```
\newcommand{\bonjour}{bonjour} % définition
```

d'origine

```
\renewcommand{\bonjour}{Bonjour !} % réécrit la définition de la commande
```

6.5 Logiciel comme générateur de code LaTeX

L'utilisation de logiciels qui génèrent du code \LaTeX est une idée permettant de reléguer le côté fastidieux au second plan. Le mode d'action est relativement simple :

1. Choisir un langage que vous maîtrisez un peu (Python, javascript...)
2. Lister les morceaux de code \LaTeX à produire. Plus vous avez d'exemples, mieux c'est.
3. Produire éventuellement un modèle \LaTeX que le logiciel viendra **modifier**.
4. Demander à l'IA de produire un script permettant de générer un document \LaTeX sur base de vos instructions.
5. Testez, débutez, modifiez et demander à l'IA d'améliorer son code grâce à vos retours.

On peut trouver beaucoup de cas d'utilisation :

Production de cartes	Gestion des modèles	Gestion d'une banque de questions	Abstraction d'exercices pour produire plusieurs versions
Produire des ressources générées procéduralement (puzzles, carrés magiques...)	Analyser du code \LaTeX et le modifier (mise à jour de fichiers anciens).	Retrouver la définition d'une commande spécifique.	Produire des rapports d'analyse d'évaluation.

Code couleur :

En cours de production par **BFcours**.

Solution disponible dans **BFcours** mais à personnaliser.

Solution disponible dans **BFcours**.

7.1 Exemple d'utilisation de Claude

La création des « TOOLBOX »

Pour fournir les kits d'outils des exercices de cette formation, j'ai demandé à Claude code de me préparer des designs dont je pourrais m'inspirer.

J'ai utilisé l'agent scripteur décrit dans cette formation.

1. Le prompt. % On ne s'est pas appliqué. C'est possible grâce à la qualité du pré-prompt.

R Dans ce répertoire pour le moment il n'y a que le fichier d'en tete et un enonce.tex vide et un commands_def.tex vide. :

```
"C:\Users\Utilisateur\Desktop\test_text\test_toolbox"
```

J'aimerais que tu me fabrique une tcolorbox nommée 'toolbox' qui me servira à donner des outils à utiliser pour résoudre un exo...
il me faut un exemple minimal de test.

contexte :

d'habitude je fais un exo ou je demande de produire du contenu via une image.
mais j'aimerais en plus donner les noms de commande et d'environnement qui son nécessaires ou peuvent être utiles pour cet exo...

le fichier d'en tete contient des commandes showcmd et showenv permettant d'afficher facilement des squelettes d'environnements et commandes. ça sera peut être utile pour l'exemple de test.

mais voila ton travail concerne le design de la toolbox, et sa présentation mise en page de manière impeccable...

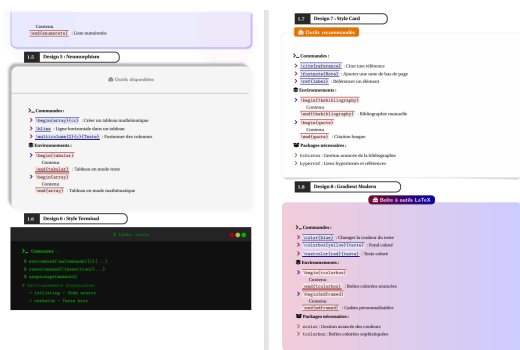
essaie plusieurs designs stylés ok ? pourquoi pas en essayant de reproduire des styles modernes de sites internet ?

fais toi plaisir dans le commands_def et dans enonce pour faire plusieurs template de ces designs. on retravaillera les meilleurs

2. L'action'. % Le modèle analyse le contenu, puis agit. Il tente même de compiler avec plusieurs compilateurs.

```
+ Le fichier commands_def.tex a été chargé et la compilation continue. 1s v
R read_output v
+ Il semble que la compilation soit bloquée ou très lente. 0s v
L list_sessions v
+ La compilation est bloquée depuis 85 secondes. 2s v
F force_terminate v
+ Je vais maintenant essayer de compiler avec lualatex qui supportera les packages emoji ... 1s v
E execute_command v
+ J'ai lancé la compilation avec lualatex. Je vais attendre un peu et vérifier la sortie. 0s v
R read_output v
+ Processus de réflexion 2s ^
La compilation avec lualatex semble progresser correctement. Je vais continuer à lire la sortie.
```

3. Le rendu. % Claude a été efficace, je n'ai plus qu'à choisir parmi les 8 exemples fournis.



4. L'investissement en temps :

En regardant les propriétés du dossier de test pour ces exemples de design, on a :

- Date de création : 8h23
- Date de fin d'édition de cet exemple : 9h

La création de cet exemple et du design des mes TOOLBOX n'a pris qu'un peu plus d'une demi-heure.

Solution de l'Exercice 2

En mathématiques, on peut `\underline{souligner}` les éléments importants, `\textbf{mettre en gras}` ou `\acc[red]{accentuer}` des mots-clés, `\emph{mettre en italique}` les théorèmes, `\hl{surligner}` - ou bien `\surligner[purple]{surligner}` - des résultats, utiliser les `\textsc{petites capitales}` ou la `{\color{green!75!black}commande}` `\frquote{voc}` pour le `\voc{vocabulaire}`, et `\fbox{encadrer}`, ou encore `\encadrer[red]{encadrer}` les formules essentielles.

Remarque(s) :

1. Les commandes « acc », « encadrer », « surligner » sont relatives au package **BFcours**.
2. La commande color doit être entourée par des crochets. Dans le cas contraire, la commande agit sur tout le paragraphe.

Solution de l'Exercice 3

1. Pour définir une commande sans paramètre qui affiche le texte « Unité non présente » en gras et en rouge :

```
\newcommand {\uniteAbsente} {}
\textcolor{red}{\textbf{Unité non présente}}
\newcommand {\uniteAbsente} {\textcolor{red} {\textbf{Unité non présente}} }
```

2. Pour définir une commande à un paramètre qui affiche « Bonjour <p> » en gras et en vert :

```
\newcommand {\bonjour} [1]{}
\textcolor{green}{\textbf{Bonjour #1}}
\newcommand {\bonjour} [1]{\textcolor{green} {\textbf{Bonjour #1}} }
```

Solution de l'Exercice 4

```
\begin{center}
```

```

\begin{minipage}{0.9\textwidth}
  \begin{center}
    \textbf{Les différents types d'alignements}
  \end{center}
  \begin{multicols}{2}
    \begin{flushleft}
      Ce texte est aligné à gauche grâce à l'environnement flushleft.
    \end{flushleft}
    \columnbreak
    \begin{flushright}
      Ce texte est aligné à droite grâce à l'environnement flushright.
    \end{flushright}
  \end{multicols}
  \begin{center}
    \begin{tcbtab}[Résumé des environnements d'alignement]{|l|c|r|}
      \hline
      flushleft & Aligne à gauche & Texte courant \\
      center & Centre le texte & Titres, équations \\
      flushright & Aligne à droite & Signature, date \\
      \hline
    \end{tcbtab}
  \end{center}
\end{minipage}
\end{center}

```

Solution de l'Exercice 5

Cette correction n'est pas implémentée.

Il s'agit d'une alternative à l'utilisation de **BFcours**.

Solution de l'Exercice 6

Il faut compiler avec **LuaLaTeX**.

Solution de l'Exercice 7

Code :

```

\begin{Definition}[Fraction]
  \begin{MultiColonnes}{3}
    \tcbitem[raster multicolumn=2] Soient deux nombres  $n$  et  $d$ . Le
    \red{quotient de  $n$  par  $d$ } , est le \red{résultat} de la \voc{division}
    du nombre  $n$  par le nombre  $d$ . Il est noté en \voc{écriture fractionnaire}
     $\encadrer[red]{\dfrac{n}{d}}$  $. \Si les nombres  $n$  et  $d$  sont
    \acc{entiers} , \Alors le nombre  $\dfrac{n}{d}$  $ est appelé une \voc{fraction}
    \tcbitem[valign=center] \bclampe Le \frquote{trait de fraction} représente une
    \acc{opération de division} .
  \end{MultiColonnes}
\end{Definition}

```

Définition

Fraction

Soient deux nombres n et d .

Le **quotient de n par d** , est le **résultat** de la **division** du nombre n par le nombre d .

Il est noté en **écriture fractionnaire** $\frac{n}{d}$.



Le « trait de fraction » représente une **opération de division**.

Si les nombres n et d sont **entiers**, **Alors** le nombre $\frac{n}{d}$ est appelé une **fraction**.

Solution de l'Exercice 8

Exemple générique de structure attendue. Il s'agit simplement de se familiariser avec les environnements disponibles.

Définition

Fonction dérivable

Une fonction f est **dérivable** en un point a si la limite suivante existe :

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$$

Exemple(s)

Calcul de dérivées

Pour $f(x) = x^2$, calculons $f'(x)$:

$$\begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} \\ &= \lim_{h \rightarrow 0} \frac{x^2 + 2xh + h^2 - x^2}{h} \\ &= \lim_{h \rightarrow 0} \frac{2xh + h^2}{h} \\ &= \lim_{h \rightarrow 0} (2x + h) \\ &= 2x \end{aligned}$$

Pour $g(x) = \sin(x)$, calculons $g'(x)$:

$$\begin{aligned} g'(x) &= \lim_{h \rightarrow 0} \frac{\sin(x+h) - \sin(x)}{h} \\ &= \lim_{h \rightarrow 0} \frac{\sin(x)\cos(h) + \cos(x)\sin(h) - \sin(x)}{h} \\ &= \sin(x) \lim_{h \rightarrow 0} \frac{\cos(h) - 1}{h} + \cos(x) \lim_{h \rightarrow 0} \frac{\sin(h)}{h} \\ &= \sin(x) \cdot 0 + \cos(x) \cdot 1 \\ &= \cos(x) \end{aligned}$$

■ Exercice 1 – Structure du document



/ 0

Calculer la dérivée de $f(x) = x^3 + 2x - 1$:

$$f'(x) = 3x^2 + 2$$

ENV-2

Solution de l'Exercice 9

`\begin{tcolorbox}[blankest]`

```

\begin{tcbraster}
[% Options du raster
  raster columns=3,% Trois colonnes
  size=fbox,% Style compact pour les boites
  raster equal height=rows,% Même hauteur par ligne
  raster width=0.99\textwidth % Taille,
  raster column skip=1pt,% Marge entre chaque colonne
  raster row skip=1pt% Marge entre chaque ligne
]
\begin{bfbox}[title=Produit remarquable]
\begin{center}

$$(a+b)^2 = a^2 + 2ab + b^2$$

\end{center}
\end{bfbox}
\end{tcbraster}
\begin{tcolorbox}[title=Identité trigonométrique]
\begin{center}

$$\sin^2(\alpha) + \cos^2(\alpha) = 1$$

\end{center}
\end{tcolorbox}
\begin{tcolorbox}[title=dérivée d'un produit]
\begin{center}

$$(u \times v)' = u' \times v + u \times v'$$

\end{center}
\end{tcolorbox}
\end{tcolorbox}

```