

# Présentation du document

## I. Mise en forme de base

### A. L'apprentissage par la pratique

Recopier et compiler le *code source* suivant :

Quelques mises en forme

```

1 \documentclass[12pt,french]{article}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{kpfonts}
5 \usepackage{mathtools,amssymb}
6 \usepackage{babel}
7
8
9 \begin{document}
10 Les consignes suivantes sont très \textbf{importantes.}\par
11 Lisez-les avec attention : \par
12 {\bfseries
13 Toutes les étapes de calculs doivent être détaillées.\par
14 Un résultat non justifié ne rapporte aucun point.
15 }
16
17 Le \fbox{barème} n'est donné qu'à titre \textit{indicatif}.
18
19 Voilà une \underline{citation} de \textsc{César} : {\itshape Alea jacta est}.
20 \end{document}
```

Code II.1

- 1°) Nommer cinq différentes mises en forme utilisées dans ce document.
- 2°) Quelle(s) commandes permettent d'obtenir ces mises en forme ?
- 3°) Quelle(s) différence(s) y a-t-il entre les deux commandes qui permettent de mettre du texte en gras ?



D'un point de vue typographique, le soulignement ne devrait jamais être utilisé. Pour mettre un texte en évidence, il doit être composé en italique. Le soulignement est réservé pour les documents manuscrits.

### B. Police et fontes

Une *police* se décline en trois caractéristiques : famille, formes et graisses qui constituent alors un ensemble de *fontes* de cette police. Le tableau ci-dessous résume les commandes permettant d'utiliser une de ces fontes.



Le caractère □ indique qu'il faut laisser un espace dans le *code source*.

		Portée		Signification des radicaux
		locale	semi-globale	
Familles	romain (par défaut)	<code>\textrm{&lt;texte&gt;}</code>	<code>\rmfamily□&lt;texte&gt;</code>	rm = roman
	sans empattement	<code>\textsf{&lt;texte&gt;}</code>	<code>\sffamily□&lt;texte&gt;</code>	sf = sans serif
	à chasse fixe	<code>\texttt{&lt;texte&gt;}</code>	<code>\ttfamily□&lt;texte&gt;</code>	tt = teletype
Formes	droit (par défaut)	<code>\textup{&lt;texte&gt;}</code>	<code>\upshape□&lt;texte&gt;</code>	up = upright (droit)
	<i>incliné</i>	<code>\textsl{&lt;texte&gt;}</code>	<code>\slshape□&lt;texte&gt;</code>	sl = slanted (penché)
	<i>italique</i>	<code>\textit{&lt;texte&gt;}</code>	<code>\itshape□&lt;texte&gt;</code>	it = italique
	PETITES CAPITALS	<code>\textsc{&lt;texte&gt;}</code>	<code>\scshape□&lt;texte&gt;</code>	sc = small caps
Graisses	médium (par défaut)	<code>\textmd{&lt;texte&gt;}</code>	<code>\mdseries□&lt;texte&gt;</code>	ms = medium
	<b>gras</b>	<code>\textbf{&lt;texte&gt;}</code>	<code>\bfseries□&lt;texte&gt;</code>	bf = bold face (gras)



Plusieurs commandes peuvent être utilisées conjointement. Par exemple pour obtenir du texte en **gras, italique et sans empattement**, on écrira :  
`\textsf{\textbf{\textit{<texte>}}}`.  
 L'ordre des commandes n'a pas d'importance mais il faut faire attention à avoir le bon nombre de paires d'accollades.

## C. Changement de la taille des fontes

La taille des fontes peut être fixée de manière absolue dans le préambule, en option à `\documentclass`. Les options disponibles sont 10pt (valeur par défaut si rien n'est indiqué), 11pt et 12pt.

Une fois définie cette taille absolue, on peut agrandir et réduire la taille d'une partie du document en utilisant des *commandes semi-globales* qui modifient alors le texte de façon relative. Le changement dépendra en effet de la taille absolue. Ces commandes sont les suivantes :

Commande	Signification et test
<code>\tiny&lt;texte&gt;</code>	minuscule
<code>\scriptsize&lt;texte&gt;</code>	taille des indices et exposants
<code>\footnotesize&lt;texte&gt;</code>	Taille des notes de bas de pages
<code>\small&lt;texte&gt;</code>	petit
<code>\normalsize&lt;texte&gt;</code>	taille définie par l'option absolue
<code>\large&lt;texte&gt;</code>	grand
<code>\Large&lt;texte&gt;</code>	plus grand
<code>\LARGE&lt;texte&gt;</code>	encore plus grand
<code>\huge&lt;texte&gt;</code>	énorme
<code>\Huge&lt;texte&gt;</code>	encore plus énorme



Les majuscules dans le nom des commandes sont importantes. De plus, il s'agit de commandes *semi-globales* donc il faut penser à mettre des accolades englobantes si on veut modifier la taille d'une partie du texte seulement.

## D. Alignement

Par défaut, le texte est *justifié*. Cela signifie que L<sup>A</sup>T<sub>E</sub>X gère les espaces entre les mots pour que le texte soit aligné à gauche *et* à droite.

Cependant, on peut parfois avoir besoin de centrer le texte, ou bien de demander uniquement un alignement à gauche ou uniquement un alignement à droite. Pour cela, on utilise respectivement les *environnements* `center`, `flushleft`, `flushright`.

Du texte au centre.

Alignement sur la gauche.

Alignement sur la droite.

```

1 \begin{center}
2 Du texte au centre.
3 \end{center}
4 \begin{flushleft}
5 Alignement sur la gauche.
6 \end{flushleft}
7 \begin{flushright}
8 Alignement sur la droite.
9 \end{flushright}
```



`\centering` est la commande semi-globale associée à l'environnement `center`. `\raggedleft` est associé à `flushright` et `\raggedright` est associée à `flushleft`.

## E. Espaces



Les espaces écrits dans le *code source* ne sont pas identiquement restitués dans le document final après compilation. Pour cela, on parlera d'un espace dans le *fichier source* et d'une espace dans le document final.

### 1. Espaces horizontales

Nous l'avons vu précédemment, pour obtenir une espace entre deux mots, il suffit de saisir un espace dans le *code source* à l'aide de la barre d'espace du clavier. Cependant, saisir plusieurs espaces ne changera rien et lors de la compilation, ils seront interprétés comme un seul et même espace. De même pour un changement de ligne (sans ligne vide !) :

Du texte sur une seule ligne.

```
1 Du_texte
2 sur_une
3 seule_ligne.
```



Le symbole `~` permet d'obtenir une *espace insécable*. En fin de ligne notamment, il faudra donc écrire `Louis~XVI` si on veut éviter que Louis soit inscrit en bout de ligne et XVI au début de la ligne suivante.

Parfois, on peut avoir besoin d'une espace horizontale ayant une longueur bien précise. Cela est possible à l'aide de la commande `\hspace{⟨longueur⟩}`. L'argument `⟨longueur⟩` est spécifié à l'aide d'un nombre suivi de son unité (sans espace entre les deux). L'unité peut être `cm`, `mm` ou bien encore `pt` mais bien d'autres aussi.

De plus, la commande `\hfill` est un espace élastique. Voilà une façon de se servir de ces deux commandes :

Les consignes sont vraiment importantes.

Les consignes sont vitales !

Inutile

**Exercice 1**

(2 points)

```
1 Les consignes sont vraiment importantes.\par
2 Les consignes sont \hspace{1.4cm} vitales !\par
3 Inutile \hspace{-1.3cm} xxxxxx\par
4 \textbf{Exercice 1} \hfill \textit{(2 points)}
```

## 2. Espaces verticales

Nous l'avons vu précédemment, pour changer de paragraphe, il suffit de saisir laisser une ligne vide dans le *code source*. La commande `\par` assure la même fonction. Cependant, plusieurs lignes vides seront toujours interprétées comme un seul changement de paragraphe, de même que la succession de plusieurs commandes `\par`.

Comment faire alors apparaître dans le document final des espaces entre deux paragraphes ?

La commande `\vspace{⟨longueur⟩}` est une solution et cela fonctionne comme pour les espaces horizontales. Cependant, les commandes `\smallskip`, `\medskip` et `\bigskip` sont simples et rapides à utiliser.

Consigne importante.

Espace standard.

Petite espace.

Espace moyenne.

Grande espace.

Espace personnelle.

```
1 Consigne importante.
2
3 Espace standard.\smallskip
4
5
6
7 Petite espace.\medskip
8
9 Espace moyenne.\bigskip
10
11 Grande espace.\par\vspace{1cm}
12 Espace personnelle.
```



La commande `\vfill` permet de créer une espace verticale élastique. Essayer de compiler l'exemple ci-dessous.

```
1 Le devoir est sur 20 points.\par\vfill
2 Tourner la page.
```

## II. Couleur



Difficile de parler de couleurs sur des photocopies noir et blanc donc pensez à taper et compiler les exemples proposés.

Afin de colorer un document, on utilise le *package* `xcolor`, chargé dans le préambule à l'aide de la commande `\usepackage{xcolor}`. `xcolor` permet d'accéder aux couleurs suivantes :

red	magenta	gray	white	violet	olive	
blue	cyan	lightgray	black	purple	teal	brown
green	yellow	darkgray	orange	pink	lime	

Là encore, il existe une commande locale et une commande semi-globale dont voilà un exemple :

Les consignes suivantes sont **importantes**.

Lisez-les avec *attention*.

Sinon, vous affronterez ma **FUREUR**.

```
1 \color{blue}
2 Les consignes suivantes sont
3 \textcolor{red}{importantes.}\par
4 Lisez-les avec \textit{attention}.\par
5 Sinon, vous affronterez ma
6 \textcolor{purple}{\textbf{\textsc{fureur}}}.
```



Le *package* `xcolor` possède différentes options qui permettent d'accéder à bien d'autres couleurs. C'est le cas de l'option `dvipsnames` qui donne accès à 68 couleurs en plus de celles de base. On écrira alors : `\usepackage[dvipsnames]{xcolor}` dans le préambule. La page 38 de la documentation du *package* permet d'en savoir davantage. Il suffit de taper sur un moteur de recherche LaTeX `xcolor doc` pour obtenir ce que l'on cherche.

Voilà un autre exemple qui montre comment faire des encadrements colorés. Quelles sont les différentes commandes ? Comment fonctionnent-elles ?

### Chapitre 1 :

*L'art de faire des encadrements*

#### I. Partie 1

```
1 \begin{center}
2   \colorbox{yellow}{\textbf{Chapitre 1 :}}\par
3   \textit{L'art de faire des encadrements}
4 \end{center}
5 \fcolorbox{red}{lightgray}{\textbf{I. Partie 1}}
```

Pour finir sur ce thème, voici la commande `\pagecolor{couleur}` qui permet de colorer le fond d'une page. Très utile pour créer un document destiné à être vidéoprojeté. En effet, le fond blanc d'un document projeté peut être fatigant pour les yeux des lecteurs. Allez-y : essayez !

## III. Mise en page

### A. Dimensions de la page

Par défaut, les dimensions de la page sont réglées en fonction de la *classe* du document.

Le *package* `geometry` est utilisé pour régler la géométrie de la page indépendamment du choix de la *classe* : dimensions du papier, orientation (portrait, paysage), dimensions des marges, particularités d'un document recto-verso, dimensions des en-têtes et pieds-de-pages...

Pour cela, on peut charger le *package* avec toute une liste d'options séparées par une virgule :

```
\usepackage[a4paper,margin=2cm]{geometry}.
```

Il est également possible de charger le *package* tout seul puis d'utiliser la commande `\geometry` qui prend en argument la même liste d'options. Ainsi, on peut également écrire :

```
\usepackage{a4paper,geometry}
\geometry{margin=2cm}
```

La documentation du *package* `geometry` liste l'ensemble des options disponibles dont voici les plus courantes ( $\langle dim \rangle$  est un nombre avec une unité de longueur) :

- \* `landscape` : orientation paysage ;
- \* `twoside` : document recto-verso ;
- \* `width=\langle dim \rangle` et `height=\langle dim \rangle` : largeur et hauteur de la page. On peut aussi utiliser `a4paper` ou `a5paper` (formats disponibles de A0 jusqu'à A6) ;
- \* `textwidth=\langle dim \rangle` et `textheight=\langle dim \rangle` : largeur et hauteur attribuée au texte ;
- \* `lmargin=\langle dim \rangle` et `rmargin=\langle dim \rangle` : respectivement marges intérieures (ou gauche) et extérieures (ou droite) ;
- \* `tmargin=\langle dim \rangle` et `bmargin=\langle dim \rangle` : respectivement marges de tête (t comme top) et de pied (b comme bottom) ;
- \* `margin=\langle dim \rangle` : fixe les quatre marges précédentes avec la même longueur.

## B. Multicolonnages

Pour écrire une partie d'un document sur deux ou plusieurs colonnes, on a recourt au *package* `multicol` qui nous permet alors d'accéder à l'*environnement* `multicols`.



Attention, le nom du package ne prend pas de S final alors que le nom de l'environnement en prend un.

Voilà deux exemples d'utilisation :

Les policiers semblent  
avoir mis la main sur  
les suspects qui ne

courraient visiblement  
pas assez vite.

```
1 \setlength{\columnseprule}{0.4mm}
2 \begin{multicols}{2}
3   Les policiers semblent avoir mis la main
4   sur les suspects qui ne courraient
5   visiblement pas assez vite.
6 \end{multicols}
```

### Formation

Ce stage L<sup>A</sup>T<sub>E</sub>X est incroyable. | Le prochain a lieu quel jour ? | Ce document est super !

```
1 \setlength{\columnseprule}{0.4pt}
2 \begin{multicols}{3}[\textbf{Formation}]
3   Ce stage \LaTeX{} est incroyable.
4   Le prochain a lieu quel jour ? Ce
5   document est super !
6 \end{multicols}
```



Pour changer de colonne à un point précis, on peut utiliser la commande `\columnbreak`.

## IV. Structurer un document

### A. Listes structurées

L<sup>A</sup>T<sub>E</sub>X gère par défaut trois types de listes :

- les listes d'énumération avec une liste d'*item* comme celle que vous être en train de lire ;
- les listes numérotées dont chaque élément est numéroté ;
- les listes de description dont chaque élément est introduit par l'objet que l'on souhaite décrire.

Voilà ce que donne la liste précédente avec les deux autres types de listes :

1. les listes d'énumération avec une liste d'*item* ;
2. les listes numérotées dont chaque élément est numéroté comme celle que vous être en train de lire ;
3. les listes de description dont chaque élément est introduit par l'objet que l'on souhaite décrire.

**les listes d'énumération** avec une liste d'*item* ;

**les listes numérotées** dont chaque élément est numéroté ;

**les listes de description** dont chaque élément est introduit par l'objet que l'on souhaite décrire comme celle que vous être en train de lire.

Toutes ces listes peuvent s'imbriquer les unes dans les autres en mélangeant ou non les différents types. On utilisera avantageusement les tabulations pour une présentation claire du *code source*.

- du pain ;
- du beurre ;
- de la confiture.

```
1 \begin{itemize}
2   \item du pain ;
3   \item du beurre ;
4   \item de la confiture.
5 \end{itemize}
```

1. Qu'est ce qu'un polygone ?
2. Qu'est ce qu'un plg ?
  - (a) Qu'est ce qu'un rectangle ?
  - (b) Qu'est ce qu'un losange ?

```
1 \begin{enumerate}
2   \item Qu'est ce qu'un polygone ?
3   \item Qu'est ce qu'un plg ?
4     \begin{enumerate}
5       \item Qu'est ce qu'un rectangle ?
6       \item Qu'est ce qu'un losange ?
7     \end{enumerate}
8 \end{enumerate}
```

**Rectangle :** voici un long texte dans lequel on parle du rectangle.

**Losange :** voici un long texte dans lequel on parle du losange.

On remarque la mise en page automatique de ce type de liste au niveau des espaces.

```
1 \begin{description}
2   \item[Rectangle :] voici un long texte dans
3     lequel on parle du rectangle.
4   \item[Losange :] voici un long texte dans lequel
5     on parle du losange.
6 \end{description}
7 On remarque la mise en page automatique
8 de ce type de liste au niveau des espaces.
```



Le *package* `enumitem` permet de personnaliser la présentation des ces différents types de listes mais également de créer de nouvelles listes. De plus, il permet de reprendre la numérotation d'une liste `enumerate` qui a été interrompue. La lecture de la documentation de ce *package* est vivement conseillée.

## B. Sectionnement

Les commandes de *sectionnement* permettent d'établir le plan du document. Les commandes les plus fréquemment utilisées sont :

```
\part[<titre court>]{<Titre>}
\chapter[<titre court>]{<Titre>}
\section[<titre court>]{<Titre>}
\subsection[<titre court>]{<Titre>}
\subsubsection[<titre court>]{<Titre>}
```



La commande `\chapter` n'existe pas dans la classe `article`. Le `<titre court>` est optionnel et permet d'afficher un titre différent dans la table des matières ou dans les en-têtes.

Recopier le code suivant et observer le résultat de la compilation :

### Commandes de sectionnement

```
1 \documentclass[12pt,french]{book}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{kpfonts}
5 \usepackage{mathtools,amssymb}
6 \usepackage{babel}
7
8 \begin{document}
9 \part{Analyse}
10
11 \chapter{Les suites}
12
13 \section{Introduction}
14
15 \section{Limite d'une suite}
16 \subsection{Limite infinie}
17 \subsection{Limite finie}
18
19 \section{Suites géométriques}
20 \subsection{Limite}
21 \subsection{Somme des termes}
22 \end{document}
```

### Code II.2



Si on ne souhaite pas de numérotation à un endroit, on peut ajouter `*`. On écrira par exemple : `\section*{Introduction}` à la place de `\section{Introduction}`.

On peut modifier la mise en forme de la numérotation des commandes `\section` et `\subsection` de la façon suivante :

```
1 \renewcommand{\thesection}{\Roman{section}.}
2 \renewcommand{\thesubsection}{\Alph{subsection}.}
```

Il suffit de taper les deux lignes précédentes dans le préambule et de recompiler. Faites un essai et essayer de comprendre le fonctionnement des commandes utilisées.



Pour modifier avec plus de précision les différents types de sectionnement, on pourra lire attentivement les documentations des packages sectsty et titlesec.

## C. Références croisées

Les *renvois* sont gérés automatiquement par L<sup>A</sup>T<sub>E</sub>X. L'intérêt de cela est évidemment de pouvoir modifier à loisir son document sans être obligé de se demander si telle ou telle référence a été changée de place ou de numérotation. Cela est très utile pour se reporter à une section ou bien encore à une liste.

Voilà un exemple :

### 1 Renvois

1. Qui suis-je ?
2. Qui est-il ?

Nous voyons que la question 2 n'est pas si simple (section 1 page 7)

```
1 \section{Renvois}\label{sec}
2   \begin{enumerate}
3     \item Qui suis-je ?\label{qu:je}
4     \item Qui est-il ?\label{qu:il}
5   \end{enumerate}
6 Nous voyons que la question \ref{qu:il}
7 n'est pas si simple
8 (section \ref{sec} page \pageref{sec})
```

La commande `\label{<étiquette>}` permet d'apposer une étiquette à l'élément que l'on souhaite référencer (ou qui est susceptible de l'être). Dans l'idéal, il faut essayer de choisir des étiquettes avec un nom suffisamment évocateur (ce qui n'est pas le cas de l'exemple précédent).

La commande `\ref{<étiquette>}` se place à l'endroit même où l'on souhaite faire notre référence à l'élément précédemment étiqueté. Le numéro alors affiché correspond au numéro de l'élément.

Enfin, `\pageref{<étiquette>}` indique le numéro de la page à laquelle se trouve l'élément à référencer.



Pour utiliser les références croisées, une seule compilation ne suffit pas. En effet, la première compilation permet simplement d'enregistrer les différentes étiquettes dans un fichier auxiliaire. Afin de pouvoir ensuite référencer ces étiquettes dans le texte, il faut procéder à une deuxième compilation. Sinon, on verra apparaître le symbole??.

## V. Exercices

### Exercice 1

Reconstituer toutes les commandes de sectionnement qui ont permis d'obtenir le plan de cette fiche. On commencera par `\chapter{Présentation du document}`

### Exercice 2

Écrire un *fichier source* complet permettant d'obtenir le document encadré ci-dessous.

La première ligne est centrée, écrite en bleu et en gras. Attention à l'écriture du nom de famille.

La citation est en italique.

Les marges sont toutes égales à 2,5 cm.

**Une citation d'un mathématicien : David HILBERT**

*Les mathématiques sont un jeu qu'on exerce selon des règles simples en manipulant des symboles et des concepts qui n'ont, en soi, aucune importance particulière.*

Citation vue sur internet.

### Exercice 3

Écrire un *fichier source* complet permettant d'obtenir le document encadré ci-dessous.

Le titre a une taille plus grande que la taille du document et les consignes ont, quant à elles, une taille plus petite. Les références de la question 3 sont générées de façon automatique.

#### CONTRÔLE DE MATHÉMATIQUES

Les consignes suivantes sont importantes :

- il faut répondre aux questions par des phrases complètes ;
- le soin de la copie et la rédaction seront pris en compte dans l'appréciation de la copie.

*Exercice 1 :*

(5 points)

1. Donner la définition d'un quadrilatère.
2. Donner la définition d'un parallélogramme.
  - (a) Quelle propriété permet de dire qu'un parallélogramme est un rectangle ?
  - (b) Quelle propriété permet de dire qu'un parallélogramme est un losange ?
3. À partir des questions 2a et 2b, déterminer une propriété permettant de dire qu'un parallélogramme est un carré.