

CONSEILS POUR BIEN TAPER UN DOCUMENT AVEC LATEX

LM204

2009-2010

§ 1. Erreurs qui ne doivent pas figurer dans un document

La présence d'une des erreurs suivantes dans un document que vous me rendrez sera sévèrement sanctionnée.

1. — *Mode mathématique.* Ne pas utiliser `$$...$$` pour entrer en mode mathématique. Utiliser `\[...\]` à la place.

incorrect	correct
Posons <code>\$\$f(x) = \ln(1+x)\$\$</code>	Posons <code>\[f(x) = \ln(1+x)\]</code>

Pour des équations numérotées, utiliser l'environnement `{equation}`. Pour aligner des équations, utiliser `{gather}`, `{align}`, `{multline}` ou leurs variantes étoilées (ne pas utiliser ni `{eqnarray}` ni `{array}` ni un tableau, l'espacement sera incorrect).

2. — *Passer à la ligne.* Ne jamais forcer un saut de ligne avec `\\`, `\newline` ou une commande équivalente. À la place, laisser une ligne blanche dans le fichier source.

incorrect	correct
À Rome, les pères avaient droit de vie et de mort sur leurs enfants.\\ \indent À Lacédémone, chaque père avait droit de corriger l'enfant d'un autre.	À Rome, les pères avaient droit de vie et de mort sur leurs enfants. À Lacédémone, chaque père avait droit de corriger l'enfant d'un autre.

L'utilisation de la commande `\\` doit être réservée aux tableaux, à `\substack` et aux environnements de type `{align}`, `{gather}` et `{multline}` (et à quelques autres).

3. — *Changement de style de police.* Ne pas utiliser les commandes obsolètes `\bf`, `\it` ou `\sc` (qui ne peuvent pas être cumulées). Utiliser `\bfseries`, `\itshape` ou `\scshape` à la place.

incorrect	correct
<pre>\begin{center} \bf\it Bien lire les instructions suivantes. \end{center}</pre>	<pre>\begin{center} \bfseries\itshape Bien lire les instructions suivantes. \end{center}</pre>

Voici une petite liste des commandes obsolètes qu’il ne faut pas utiliser et ce qu’il faut utiliser à la place.

incorrect	<code>\bf</code>	<code>\it</code>	<code>\rm</code>	<code>\sc</code>	<code>\sf</code>	<code>\sl</code>	<code>\tt</code>
correct	<code>\bfseries</code>	<code>\itshape</code>	<code>\rmfamily</code>	<code>\scshape</code>	<code>\sffamily</code>	<code>\slshape</code>	<code>\ttfamily</code>

4. — *Aligner des équations.* Ne pas utiliser l’environnement `{eqnarray}`. Utiliser `{align}`, `{gather}` ou `{multline}` à la place.

incorrect	correct
<pre>\begin{eqnarray} A &=& B \\\ &=& C \end{eqnarray}</pre>	<pre>\begin{align} A &=& B \\\ &=& C \end{align}</pre>

(Noter la différence dans le placement des `&` dans `{eqnarray}` et `{align}`.)

5. — *Packages de polices.* Toujours charger le package `lmodern`, même si vous changez de police après ¹. Les autres packages de police disponibles que vous êtes autorisés à utiliser pour les documents que vous me rendez sont ² :

- `\usepackage{fourier}`
- `\usepackage[utopia]{mathdesign}`
- `\usepackage[charter]{mathdesign}`
- `\usepackage[garamond]{mathdesign}`
- `\usepackage{kpfonts}`

Il y a aussi les deux packages suivants, mais il vaut mieux ne pas les utiliser s’il y a des mathématiques un peu compliquées, car les métriques de ces polices sont mal réglées.

- `\usepackage{txfonts}` (autre possibilité : `\usepackage{mathptmx}`)
- `\usepackage{pxfonts}` (autre possibilité : `\usepackage{mathpazo}`)

(Les packages `mathptmx` (bien noter le `x` à la fin) et `mathpazo` n’ont pas de problèmes de réglages, mais sont des polices beaucoup moins complètes, donc pas forcément plus recommandables.)

1. Ceci permet d’éviter d’avoir des polices de mauvaise qualité dans le PDF.

2. La police `garamond` nécessite une installation manuelle, chose qui est non exigible pour ce module.

Dans les documents à me rendre, tout autre package de police est strictement interdit ; voici une liste non exhaustive de packages à ne pas utiliser et ce qu'on peut utiliser à la place.

incorrect	correct
<code>\usepackage{pslatex}</code>	<code>\usepackage{txfonts}</code>
<code>\usepackage{times}</code>	<code>\usepackage{txfonts}</code>
<code>\usepackage{mathptm}</code>	<code>\usepackage{txfonts}</code>
<code>\usepackage{charter}</code>	<code>\usepackage[charter]{mathdesign}</code>
<code>\usepackage{bcharter}</code>	<code>\usepackage[charter]{mathdesign}</code>
<code>\usepackage{palatino}</code>	<code>\usepackage{pxfonts}</code>
<code>\usepackage{mathppl}</code>	<code>\usepackage{pxfonts}</code>

6. — *Ordre de chargement des packages.* Cette erreur-ci est beaucoup moins grave, mais il est important de toujours charger hyperref en dernier. Il peut y avoir des problèmes dans le cas contraire. Cette règle n'est pas absolue, certains packages nécessitant d'être chargés après hyperref (par exemple, le package glossaries pour faire des glossaires ou tabularx pour éviter des problèmes avec \footnote).

§ 2. Remarques sur l'utilisation de \$, \[et \]

1. — *Bien taper une formule.* Ne rentrer en mode mathématique qu'une seule fois par formule et n'en sortir qu'à la fin.

incorrect	correct
On a $A^2 = (x+y) \times (x+z)$.	On a $A^2 = (x+y) \times (x+z)$.

REMARQUE. — Ne pas mettre les majuscules entre dollars peut être vu comme un moyen d'obtenir des majuscules droites. Ce n'est pas la bonne façon de faire, il faut utiliser `\mathrm` pour cela. Bien entendu, si on veut que toutes les majuscules du document soient droites, il ne faut pas toutes les taper avec un `\mathrm`, mais plutôt dire à LaTeX de le faire automatiquement ; pour cela, on peut par exemple utiliser le package `fourier` avec l'option `upright` :

```
\usepackage[upright]{fourier}
```

2. — *Équations sur une ligne à part.* Si on veut qu'une équation soit hors-texte (c'est-à-dire mise en évidence à la ligne), il faut utiliser `\[...\]`, `{align}`, `{gather}` ou `{multline}`.

incorrect	correct
<p>Considérons la suite (u_n) définie par</p> $u_0 = 1$ $u_{n+1} = \sin u_n$ <p>Montrons que cette suite...</p>	<p>Considérons la suite (u_n) définie par</p> $\begin{gather*} u_0 = 1 \\ u_{n+1} = \sin u_n \end{gather*}$ <p>Montrons que cette suite...</p>

Si jamais on ne veut pas que les mathématiques soient centrées, on utilisera l'option `fleqn` de `\documentclass` :

```
\documentclass[fleqn]{article}
```

3. — Centrer une équation. Ne jamais centrer manuellement des mathématiques avec `{center}` ou `\centering` ou quoi que ce soit de ce style.

incorrect	correct
<p>Soit f définie par</p> $\begin{center} f(x) = \frac{1}{x} + \frac{1}{\ln(1-x)} \end{center}$ <p>Montrons que f est continue en 0.</p>	<p>Soit f définie par</p> $[f(x) = \frac{1}{x} + \frac{1}{\ln(1-x)}]$ <p>Montrons que f est continue en 0.</p>

incorrect	correct
<p>Soit f définie par</p> $\begin{center} \begin{equation} f(x) = \frac{1}{x} + \frac{1}{\ln(1-x)} \end{equation} \end{center}$ <p>Montrons que f est continue en 0.</p>	<p>Soit f définie par</p> $\begin{equation} f(x) = \frac{1}{x} + \frac{1}{\ln(1-x)} \end{equation}$ <p>Montrons que f est continue en 0.</p>

4. — Sauts de lignes entre des formules. Ne jamais sauter de lignes arbitrairement autour d'un `\[...\]`, `{align}`, `{equation}` ou `{gather}`.

incorrect	correct
<p>Soit f définie par</p> $[f(x) = \frac{1}{x} + \frac{1}{\ln(1-x)}]$ <p>Cette fonction est bornée au voisinage de 0 car $f(x) = \frac{1}{x^2} + o(x)$.</p>	<p>Soit f définie par</p> $[f(x) = \frac{1}{x} + \frac{1}{\ln(1-x)}]$ <p>Cette fonction est bornée au voisinage de 0 car $f(x) = \frac{1}{x^2} + o(x)$.</p>

On peut laisser une ligne blanche après un `\[...\]` à condition que l'on veuille changer de paragraphe (ce qui n'est pas le cas si la phrase continue après la formule).

5. — Formules multiples Ne jamais mettre plusieurs `\[...]`, `{align}`, `{equation}` ou `{gather}` les uns à la suite des autres ; utiliser plutôt un environnement `{align}` ou `{gather}` avec un `\notag` s'il y a besoin qu'une ligne soit non numérotée.

incorrect	correct
Montrons que f est bornée au voisinage de 0 : $f(x) = \frac{1}{x} + \frac{1}{x+O(x^2)}$ $f(x) = O(1)$ d'où le résultat.	Montrons que f est bornée au voisinage de 0 : $\begin{gathered} f(x) = \frac{1}{x} + \frac{1}{x+O(x^2)} \\ f(x) = O(1) \end{gathered}$ d'où le résultat.

6. — Mettre du texte dans les formules. Pour taper du texte à l'intérieur d'une formule, il faut utiliser `\text`, pas `\mathrm`, `\mbox` ou quoi que ce soit d'autre de ce genre. De plus, pour des formules hors-texte, on met généralement un `\quad` avant et après chaque commande `\text`.

incorrect	correct
$f(x) = 0 \text{ et donc } x = \pm 1$ $\sum_{x \neq 0 \text{ et } x \neq 1} f(x)$	$f(x) = 0 \quad \text{et donc} \quad x = \pm 1$ $\sum_{x \neq 0 \text{ et } x \neq 1} f(x)$

(La commande `\mbox` est particulièrement fautive car, en indice ou en exposant, le texte ne sera pas de la bonne taille.)

7. — Enchaîner des formules dans le texte. Lorsqu'on écrit des formules dans le texte, on n'utilise pas ³ `\text`, mais on sort du mode mathématique puis on y reentre à chaque fois

incorrect	correct
On a $f(x) = 0$ et donc $x = \pm 1$.	On a $f(x) = 0$ et donc $x = \pm 1$.

§ 3. Remarques sur l'utilisation des automatismes de LaTeX

1. — Ne pas numéroté à la main des théorèmes, figures, etc. LaTeX dispose de méthodes simples pour créer des éléments numérotés (ou non numérotés) générés automatiquement : il faut les utiliser.

3. Sauf si c'est justifié, mais en règle générale, ce n'est pas le cas.

incorrect	correct
<pre>\textbf{Théorème~1.} Tout entier \$\geq 2\$ est produit de nombres premiers.</pre>	<pre>\begin{theoreme} Tout entier \$\geq 2\$ est produit de nombres premiers. \end{theoreme}</pre>
<pre>\textit{Remarque.} Ce théorème est faux si \$n=2\$.</pre>	<pre>\begin{remarque} Ce théorème est faux si \$n=2\$. \end{remarque}</pre>

(Les environnements `{theoreme}` et `{remarque}` étant créés dans le préambule avec `\newtheorem` du package `amsthm`.)

incorrect	correct
<pre>\begin{center} \includegraphics{image} Figure 1 - Image d'un tigre \end{center}</pre>	<pre>\begin{center} \includegraphics{image} \captionof{figure}{Image d'un tigre} \end{center}</pre>

(La commande `\captionof` nécessite le package `caption`.) Si on veut que la figure soit placée automatiquement par LaTeX, on utilisera l'environnement `{figure}` et la commande `\caption` :

incorrect	correct
<pre>\begin{figure}\centering \includegraphics{image} Figure 1 - Image d'un tigre \end{figure}</pre>	<pre>\begin{figure}\centering \includegraphics{image} \caption{Image d'un tigre} \end{figure}</pre>

2. — Ne pas faire référence à un élément à la main. LaTeX dispose d'un mécanisme de `\label` et `\ref` qui permet de faire automatiquement référence à un numéro de section, d'équation, de théorème, de page, etc. Il est indispensable d'utiliser ces mécanismes et de ne pas taper ces numéros à la main : en cas de changement, tout est mis à jour automatiquement au bout de deux compilations.

incorrect	correct
<pre>D'après le théorème de Fermat~2.13 vu page~34, ...</pre>	<pre>D'après le théorème de Fermat~\ref{th.de.fermat} vu page~\pageref{th.de.fermat}, ...</pre>

3. — Ne pas oublier de mettre le contexte d'une référence. Si on fait une référence avec `\ref` à un théorème, une section, une figure, etc., il faut absolument mettre le mot théorème, le symbole § ou le mot figure avant le `\ref`, sinon la phrase est incompréhensible.

incorrect	correct
D'après <code>\ref{th.gauss}</code> , on a ...	D'après le théorème~ <code>\ref{th.gauss}</code> ,
D'après <code>\ref{section:resultats}</code> ,	D'après le §~ <code>\ref{section:resultats}</code> ,
Voir <code>\ref{fig:tigre}</code> .	Voir figure~ <code>\ref{fig:tigre}</code> .

4. — Références à un numéro d'équation. Pour faire référence à un numéro d'équation, il faut utiliser `\eqref` et non `\ref`.

incorrect	correct
Voir <code>formule~(\ref{der.prod})</code> .	Voir <code>formule~\eqref{der.prod}</code> .

5. — Citations bibliographiques. Il faut obligatoirement utiliser la commande `\cite`, éventuellement avec un argument optionnel pour spécifier le numéro de théorème, de page, etc.

incorrect	correct
D'après~ <code>[Lan03]</code> , ...	D'après~ <code>\cite{Lang.Algebra}</code> , ...
D'après~ <code>[Lan03, p.~46]</code> , ...	D'après~ <code>\cite[p.~46]{Lang.Algebra}</code> ,

6. — Faut-il utiliser `\itshape` ou `\textit` ? En règle générale, on utilisera les commandes à argument (comme `\textit`, `\emph`, `\textbf`, etc.) pour changer la police de quelques mots ou phrases ; pour changer la police d'un ou de plusieurs paragraphes en entiers, on utilisera plutôt les bascules (comme `\itshape`, `\em`, `\bfseries`, etc.). Cette façon de faire peut avoir son importance avec l'italique, car LaTeX ajoute automatiquement une correction optique lorsqu'on utilise les commandes à arguments (c'est donc elles qu'il faut utiliser quand on met de l'italique dans du texte normal). Comparer attentivement les deux phrases suivantes en notant la distance entre le *f* et le mot « est ».

incorrect : ce *pendentif* est joli

correct : ce *pendentif* est joli

incorrect	correct
ce <code>\itshape pendentif</code> est joli	ce <code>\textit{pendentif}</code> est joli
ce <code>\em pendentif</code> est joli	ce <code>\emph{pendentif}</code> est joli

Noter que les commandes à argument comme `\textbf` ou `\textsc` n'acceptent pas de changement de paragraphe dans leur argument ; il faut donc utiliser `\bfseries` ou `\scshape` pour mettre plusieurs paragraphes en gras ou en petites capitales.

7. — Changer le nom des éléments. Il est possible de renommer les éléments comme « chapitre », « table des matières », etc. en redéfinissant `\chaptername`, `\contentsname`, etc. Quand on utilise le package `babel`, il faut néanmoins faire attention à comment s’y prendre. Au lieu d’utiliser un `\renewcommand` après le `\begin{document}`, il vaut mieux passer par `\addto` et `\captionsfrench` (à mettre dans le préambule), la modification sera robuste à un changement de langue.

incorrect	correct
<code>\renewcommand\contentsname{Sommaire}</code>	<code>\addto\captionsfrench{ \renewcommand\contentsname{Sommaire} }</code>

8. — Placement des flottants. Un flottant est là où LaTeX choisi qu’il soit. Si on veut un flottant en début de page, on le met à l’endroit où il apparaît logiquement dans le texte (par exemple juste après le texte qui en parle), on spécifie l’option `[t]` dans `{figure}` ou `{table}` et on laisse à LaTeX le choix du placement (si on veut qu’une figure ou qu’un tableau soit à un endroit précis, on ne la (ou le) met pas dans une `{figure}` ou une `{table}` mais dans un `{center}` (par exemple) et, si on veut une légende, on utilise la commande `\captionof` du package `caption`).

§ 4. Pièges sur les commandes et environnements

1. — Espacement après une commande. Si une commande ne prend pas d’argument, elle mange l’espace qui la suit. Pour éviter ce problème, il faut rajouter une paire d’accolades vides `{ }` après la commande.

incorrect	correct
<code>C'est la \og bonne\fg méthode.</code>	<code>C'est la \og bonne\fg{} méthode.</code>
<code>\LaTeX a été crée par Lamport.</code>	<code>\LaTeX{} a été crée par Lamport.</code>
<code>Word\textregistered date de 1983.</code>	<code>Word\textregistered{} date de 1983.</code>

2. — Environnement pour le résumé d’un article. Pour faire un résumé, il faut utiliser l’environnement `{abstract}`. Ne pas utiliser la commande `\abstract`, sinon tout le reste du document se retrouvera également à l’intérieur du résumé.

incorrect	correct
<code>\abstract{Dans cet article, nous démontrons ...}</code>	<code>\begin{abstract} Dans cet article, nous démontrons ... \end{abstract}</code>

3. — Environnement et commandes pour centrer du texte. Pour centrer du texte, il faut utiliser soit l’environnement `{center}`, soit la commande `\centering`, mais pas `\center`.

incorrect	correct
<code>{\center Texte centré}</code>	<code>\begin{center}</code> Texte centré <code>\end{center}</code>

Pour centrer du texte, éviter d’utiliser `\centerline` qui ne fonctionne pas comme il faut dans certaines situations.

§ 5. Mise en forme manuelle

1. — Sauts de pages explicites. Ne pas utiliser des sauts de pages explicites comme `\newpage`, `\pagebreak`, `\clearpage` ou encore `\cleardoublepage`. En cas de changement (rajout de texte), on peut se retrouver avec une page avec quelques lignes puis un saut de page inexpliqué que l’on peut ne pas voir si le changement est fait à la dernière minute.

2. — Espaces explicites. Il est rare d’avoir vraiment besoin de mettre des espacements explicites comme `\bigskip`, `\medskip`, `\smallskip` ou `\bigbreak`, `\medbreak` et `\smallbreak`. Par exemple, au mieux de rajouter un `\smallskip` après toutes les sections, on redéfinira l’espace après les sections avec le package `titlesec`.

incorrect	correct
<code>\medskip</code> <code>\subsubsection{Introduction}</code> <code>\bigskip</code>	<code>\subsubsection{Introduction}</code>

incorrect	correct
<code>\smallskip</code> <code>\begin{definition}</code> Bla bla. <code>\end{definition}</code> <code>\bigbreak</code>	<code>\begin{definition}</code> Bla bla. <code>\end{definition}</code>

Dans certains cas de figures bien précis, il peut être judicieux d’utiliser `\smallbreak`, `\medbreak` ou `\bigbreak` (qui sont à préférer à `\smallskip`, `\medskip` ou `\bigskip` qui peuvent avoir des effets indésirables en cas de mauvais placement). Un exemple : on souhaite séparer un tableau mis dans un environnement `{center}` de ce qui le suit. Dans ce cas, on peut utiliser un `\bigbreak` car c’est pour séparer deux éléments, pas pour faire des ajustements manuels hasardeux. Un autre cas est si on veut nettement séparer

deux paragraphes entre eux (pour marquer un changement de sujet, par exemple), mais, bien sûr, cela ne doit concerner que les quelques paragraphes pour lesquels c’est justifié par le sens du texte.

3. — Lignes blanches dans le fichier source. Ne pas laisser de lignes blanches n’importe où dans le fichier source. Une ligne blanche correspond à un changement de paragraphe et ne doit pas être utilisée pour autre chose (certainement pas, par exemple, pour forcer un retour à la ligne).

incorrect	correct
<pre>Corneille a dit : \begin{quote} Que le parti plus faible obéisse au plus fort ; \end{quote} et Racine : \begin{quote} Je méditais ma fuite aux rives étrangères. \end{quote}</pre>	<pre>Corneille a dit : \begin{quote} Que le parti plus faible obéisse au plus fort ; \end{quote} et Racine : \begin{quote} Je méditais ma fuite aux rives étrangères. \end{quote}</pre>

incorrect	correct
<pre>Corneille a dit : \begin{quote} Que le parti plus faible obéisse au plus fort~; \end{quote} Passons maintenant aux citations de Racine. Il a dit : \begin{quote} Je méditais ma fuite aux rives étrangères. \end{quote}</pre>	<pre>Corneille a dit : \begin{quote} Que le parti plus faible obéisse au plus fort~; \end{quote} Passons maintenant aux citations de Racine. Il a dit : \begin{quote} Je méditais ma fuite aux rives étrangères. \end{quote}</pre>

(Dans le premier exemple, il n’y a pas de changement de paragraphe car les citations sont au milieu d’une phrase. Dans le second exemple, il y a un changement de paragraphe quand on change d’auteur : il faut donc laisser une ligne blanche dans le fichier source.)

Les exemples précédents avec `{quotation}` valent aussi pour `{quote}`, `{itemize}`, `{enumerate}`, etc. (ainsi que pour les équations hors-texte).

4. — Changements de police. Ne pas polluer les commandes ou environnements de balisage logique par des changements de polices ; la bonne méthode est de changer le style de ces commandes et environnements. Par exemple, pour obtenir « THÉORÈME »

au lieu de « **Théorème** », on ne mettra pas un `\scshape` dans le `\newtheorem`, on changera le style associé à la définition (avec la commande `\newtheoremstyle` du package `amsthm`). Autre exemple : pour mettre le titre en gras et en capitales, on tapera le titre normalement puis on utilisera le package `titling` pour afficher le titre en gras et en majuscules (avec la commande `\MakeUppercase`)⁴.

incorrect	correct
<code>\newtheorem{theo}{\scshape Théorème}</code>	<code>\newtheorem{theo}{Théorème}</code>
<code>\begin{definition}</code> <code>\itshape Bla bla bla</code> <code>\end{definition}</code>	<code>\begin{definition}</code> <code>Bla bla bla</code> <code>\end{definition}</code>
<code>\title{\bfseries LOGARITHMES}</code>	<code>\title{Logarithmes}</code>
<code>\subsubsection{\color{red}Introduction}</code>	<code>\subsubsection{Introduction}</code>

De même, ne pas abuser des changements de polices en écrivant par exemple systématiquement

L'augmentation est de `\textsf{30}~\%` et ...

Dans le texte, les changements de polices doivent être limités au maximum. Au besoin, si un changement de style de police revient souvent et a une signification logique, faire une commande personnelle pour cela.

§ 6. Mise en forme

1. — Éviter le souligné. Il y a beaucoup de meilleurs moyens que le souligné pour mettre en valeur du texte (l'italique est à préférer au gras, un peu trop voyant).

incorrect	correct
<code>Ceci est \underline{important}.</code>	<code>Ceci est \emph{important}.</code>
<code>Ceci est \ul{important}.</code>	<code>Ceci est \emph{important}.</code>

2. — Éviter le penché. Le penché (`\textsl` ou `\slshape`) ne devrait jamais être utilisé. Préférer le véritable italique (`\textit`, `\itshape` ou `\emph` et `\em`).

incorrect	correct
<code>Ceci est \textsl{important}.</code>	<code>Ceci est \emph{important}.</code>

4. On pourra consulter le document http://www.math.jussieu.fr/~goutet/latex/seance_6/seance_6.pdf pour ce genre de personnalisations ; c'est en dehors du programme du module par correspondance (CNED).

3. — *Choix des couleurs.* Éviter les couleurs agressives (vert fluo, jaune vif, etc.) ainsi que les couleurs trop pâles (vert tamisé à 20 %, etc.).

incorrect	correct
<code>\textcolor{green}{mot}</code>	<code>\textcolor{green!60!black}{mot}</code>
<code>\textcolor{green!20}{mot}</code>	<code>\textcolor{green!60!black}{mot}</code>
<code>\textcolor{cyan}{mot}</code>	<code>\textcolor{blue}{mot}</code>

4. — *Mélanges de polices.* Éviter de mélanger trop de polices dans un même document. Le maximum devrait être : une police pour le texte, une pour le titre, une pour les sections et une pour les algorithmes et autres codes informatiques (police monospace). Une chose à ne pas faire est d'utiliser une police différente par niveau de sectionnement.

§ 7. Formattage des mathématiques

1. — *Oubli des dollars.* Toute formule mathématique dans le texte doit être entre dollars (ou entre `\(...\)`). Toute formule hors-texte doit être entre `\[...\]` (ou dans un environnement `{equation}`, `{align}`, `{gather}` ou encore `{multline}`).

incorrect	correct
Soit M le milieu du segment [AB].	Soit M le milieu du segment $[AB]$.
Posons	Posons
$f(x) = \ln(1+x)$	$f(x) = \ln(1+x)$

2. — *Utilisation de `\mathit` ou `\textit`.* Les commandes `\mathit` et `\textit` ne doivent pas être utilisées à tort et à travers⁵ ; en règle générale, il suffit de mettre les formules entre dollars pour obtenir ce qu'il faut.

incorrect	correct
Lorsque $\textit{n} = 3$, on a $\mathit{N} = 8$.	Lorsque $n = 3$, on a $N = 8$.

3. — *Confusions entre `\mathrm` et `\text`.* On utilise uniquement `\mathrm` pour mettre des mathématiques en romain (exemple typique : X_{\max} s'écrit X_{max}), pas pour taper du texte.

incorrect	correct
$x^2 = 1 \quad \mathrm{et~donc}$	$x^2 = 1 \quad \text{et donc}$
$x = \pm 1$	$x = \pm 1$

5. La commande `\textit` ne devrait pratiquement jamais être utilisée et la commande `\mathit` doit être réservée aux « sigles » en lettres capitales italiques comme dans $PGL_n(\mathbb{C})$ ou $SO_n(\mathbb{R})$.

4. — Utilisation de `\mathrm` pour les fonctions. On n'utilise jamais `\mathrm` dans les formules pour obtenir des fonctions du type `grad`, `div`, `pgcd`, `ppcm`, etc. En lieu et place, on utilise, dans le préambule, la commande `\DeclareMathOperator` du package `amsmath` pour créer une nouvelle commande adéquate.

incorrect	correct
<code>\[\mathrm{pgcd}(a,b) \cdot \mathrm{ppcm}(a,b) = ab\]</code>	<code>\[\pgcd(a,b) \cdot \ppcm(a,b) = ab\]</code>

où l'on a rajouté, dans le préambule,

```
\DeclareMathOperator{\pgcd}{pgcd}
\DeclareMathOperator{\ppcm}{ppcm}
```

5. — Obtenir plusieurs lignes d'indices. Si on a besoin de plusieurs lignes d'indice, on utilisera la commande `\substack` ; ne pas utiliser `\atop` ni `\stackrel` pour faire ce genre de choses, ils ne sont pas fait pour cela et ne donne pas le bon résultat sans ajustements manuels envahissants.

incorrect	correct
<code>\[\sum_{i,j \atop i+j=n} C_n^i\]</code>	<code>\[\sum_{\substack{i,j\\i+j=n}} C_n^i\]</code>
<code>\[\sum_{\stackrel{i,j}{i+j=n}} C_n^i\]</code>	<code>\[\sum_{\substack{i,j\\i+j=n}} C_n^i\]</code>

6. — Mettre une formule au-dessus ou au-dessous d'un symbole. Le package `amsmath` dispose de commandes `\underset` et `\overset` qui fonctionnent correctement dans tous les cas de figures pour placer une formule au-dessus ou en-dessous d'un symbole. On utilisera donc uniquement ces commandes pour faire ce genre de choses ⁶.

$$x_n \underset{n \rightarrow 0}{=} n \qquad u_n \overset{\text{def}}{=} f(n)$$

incorrect	correct
<code>\[x_n \mathop{=}\limits_{n \rightarrow 0} n\]</code>	<code>\[x_n \underset{n \rightarrow 0}{=} n\]</code>
<code>\[u_n \stackrel{\text{def}}{=} f(n)\]</code>	<code>\[u_n \overset{\text{def}}{=} f(n)\]</code>

7. — Ajuster manuellement l'espacement avec `\,`. La commande `\,` (ainsi que `\;` ou `\:`) ne doit être utilisée que pour corriger un espacement défectueux, donc relativement rarement. Elle ne doit surtout pas être utilisée de manière systématique à chaque fois qu'on tape une certaine expression.

6. La commande `\stackrel` est moins performante, donc peut être considérée obsolète et ne devrait plus être utilisée ; de même, ne pas utiliser des astuces basées sur `\mathop` et `\limits` qui ont l'inconvénient de modifier l'espacement.

incorrect	correct
<code>\[2\,\mathrm{ln}\,2=\mathrm{ln}\,4\]</code>	<code>\[2 \ln 2 = \ln 4\]</code>

Il y a un certain nombre de cas où il faut rajouter des espacements manuels, mais dans ce cas il vaut mieux définir une commande (à mettre dans le préambule) où cet ajustement est fait ; ceci permet d'assurer une homogénéité au document. C'est par exemple le cas des éléments différentiels dt des intégrales qui doivent en général être précédés d'un petit espacement. Au lieu d'écrire systématiquement `\,\mathrm{d}`, il faut définir une commande personnelle pour s'en occuper. On verra comment bien définir cette commande au § 9, n° 3, p. 21. Une fois que cette commande a été définie, il faut l'utiliser à chaque fois pour assurer l'homogénéité du document :

incorrect	correct
<code>\int_a^b f(t)\,\mathrm{d}t</code>	<code>\int_a^b f(t) \diff t</code>

Un autre cas de figure où il faut faire l'ajustement est pour le `\middle|` dans une construction ensembliste. Là aussi, il faut définir une commande personnelle judicieuse (voir au § 9, n° 6, p. 22).

incorrect	correct
<code>\left\{x\in X,\middle \,f(x)=0\right\}</code>	<code>\enstq{x\in X}{f(x)=0}</code>

Un autre cas est devant les points-virgules dans les intervalles (lorsqu'il ne sont pas en indice ou exposant). Comme pour les deux autres cas de figures précédents, on définira une commande personnelle pour les intervalles (voir § 9, n° 1, p. 20) puis on l'utilisera partout.

incorrect	correct
<code>\$[a\,;\,b]\$\</code>	<code>\$\$\intervalleff{a}{b}\$\$</code>

§ 8. Taper les formules

1. — Oubli des accolades en exposant. Quand il y a plusieurs caractères en exposant, les accolades sont indispensables pour ne pas obtenir x^14 au lieu de x^{14} .

incorrect	correct
<code>\$x^14\$</code>	<code>\$x^{14}\$</code>

2. — Confusions entre les deux types de virgules. La virgule a deux rôles : un rôle de séparateur d'arguments ou de coordonnées (comme dans $f(x, y)$) et un rôle de séparateur décimal (comme dans $\pi \approx 3,14159$). Pour le séparateur décimal, il faut utiliser $\{, \}$ afin qu'il n'y ait pas d'espace parasite après la virgule. Voici des exemples sur les formules suivantes⁷.

$$M_0(x_0, y_0) \quad M(\tfrac{3}{2}, 1) = M(1,5, 1) \quad \sqrt{2} \approx 1,4142$$

incorrect	correct
$\$M_0(x_0\{,\}y_0)\$$	$\$M_0(x_0,y_0)\$$
$\$M(\tfrac{3}{2},1) = M(1,5,1)\$$	$\$M(\tfrac{3}{2},1) = M(1\{,\}5,1)\$$
$\$\sqrt{2} \approx 1,4142\$$	$\$\sqrt{2} \approx 1\{,\}4142\$$

3. — Confusions entre |, \lvert, \rvert et \mid. La barre verticale $|$ a quatre utilisations distinctes : restriction, valeur absolue ouvrante, valeur absolue fermante et divisibilité. Il faut utiliser une commande spécifique pour chacune de ces occasions, sinon l'espacement ne sera pas correct. Voici des exemples sur les formules suivantes⁸.

$$|-x| = |x| \quad d \mid n \quad \{x \in X \mid f(x) > 0\} \quad f|_X$$

incorrect	correct
$\$ -x = x \$$	$\$\lvert -x \rvert = \lvert x \rvert\$$
$\$d \mid n\$$	$\$d \mid n\$$
$\$\{x \in X \mid f(x) > 0\}\$$	$\$\{x \in X \mid f(x) > 0\}\$$
$\$f \mid_X\$$	$\$f _X\$$

Bien entendu, pour toutes ces commandes, le mieux est de définir des commandes personnelles : une commande `\abs` pour les valeurs absolues (voir § 9, n° 4, p. 21), une commande `\divide` pour la divisibilité (facultatif) et une commande `\ens tq` pour la construction ensembliste (voir § 9, n° 6, p. 22).

4. — Confusions entre \Vert, \lVert, \rVert et \parallel. La barre verticale double \parallel a trois utilisations distinctes : norme ouvrante, norme fermante et divisibilité exacte (ainsi que parallélisme). Il faut utiliser une commande spécifique pour chacune de ces occasions, sinon l'espacement ne sera pas correct. Voici des exemples sur les formules suivantes.

$$\|\vec{u}\| = 1 \quad \left\| \frac{\vec{u}}{\lambda} \right\| = 1 \quad p^3 \parallel n \quad (AB) \parallel (CD)$$

7. En fait, pour éviter toute ambiguïté dans la formule $M(1,5, 1)$, il vaudrait mieux utiliser le point-virgule pour séparer les deux coordonnées : $M(1;5; 1)$.

8. Pour la restriction, utiliser $|$ ne donnera pas toujours le bon espacement, voir § 9, n° 8, p. 22 pour la bonne façon de faire.

incorrect	correct
$ \vec{u} = 1$	$\lVert\vec{u}\rVert = 1$
$ \frac{\vec{u}}{\lambda} = 1$	$\left\lVert \frac{\vec{u}}{\lambda} \right\rVert = 1$
$p^3 \text{ \texttt{\textbackslash Vert} } n$	$p^3 \text{ \texttt{\textbackslash parallel} } n$
$(AB) \text{ \texttt{\textbackslash\textbackslash} } (CD)$	$(AB) \text{ \texttt{\textbackslash parallel} } (CD)$

Bien entendu, pour toutes ces commandes, le mieux est de définir des commandes personnelles : une commande `\norme` pour les normes (voir § 9, n° 4, p. 21) et une commande `\diviseexactement` pour la divisibilité exacte (facultatif).

5. — Confusions entre `:` et `\colon`. Le deux-points a deux significations différentes : l’une comme ponctuation (auquel cas, il faut utiliser `\colon`) et l’autre comme opérateur de relation⁹ (auquel cas il faut utiliser `:`). Voici des exemples sur les formules suivantes.

$$f: X \rightarrow Y \quad \{x : \sin x = 0\} \quad A := 3$$

incorrect	correct
$f : X \text{ \texttt{\textbackslash to} } Y$	$f \text{ \texttt{\textbackslash colon} } X \text{ \texttt{\textbackslash to} } Y$
$\{x \text{ \texttt{\textbackslash colon} } \sin x = 0\}$	$\{x : \sin x = 0\}$
$A := 3$	$A \text{ \texttt{\textbackslash coloneq} } 3$

(La commande `\coloneqq` nécessite le package `mathtools`.)

6. — Confusions entre `< ... >` et `\langle ... \rangle`. Il ne faut pas confondre `<` et `>` (« strictement plus petit » et « strictement plus grand ») avec les crochets angulaires `\langle` et `\rangle`. Voici des exemples sur les formules suivantes.

$$a^2 - b^2 < a^2 + b^2 \quad \langle \vec{u}, \vec{v} \rangle$$

incorrect	correct
$a^2 - b^2 \text{ \texttt{\textbackslash} } < a^2 + b^2$	$a^2 - b^2 < a^2 + b^2$
$\langle \vec{u}, \vec{v} \rangle = 0$	$\langle \vec{u}, \vec{v} \rangle = 0$

(Si vous utilisez une version de TeXmaker antérieure à la 2.3, faire bien attention, le symbole inséré si on utilise la barre latérale n’est pas `<` mais `\<` (qui n’est pas la bonne commande). Il ne faut donc *pas* utiliser cette barre latérale pour les deux symboles `<` et `>`.)

9. Un opérateur de relation est un objet du même type que `=`, `∈`, `<`, `\mid` ou `\parallel`.

7. — *Nature fermante ou ouvrante de certains délimiteurs.* En typographie française, le crochet ouvrant [peut être utilisé pour fermer une expression et vice-versa. Il faut donc, dans certains cas, forcer le type ouvrant ou fermant, notamment pour les intervalles. Voici des exemples sur les formules suivantes.

$$[a ; b] \quad]a ; b] \quad [a ; b[\quad]a ; b[$$

incorrect	correct
<code>\$[a;b]\$\</code>	<code>\$[a;b]\$\</code>
<code>\$]a;b]\$\</code>	<code>\$\mathopen{]}a;b]\$\</code>
<code>\$[a;b[\$\</code>	<code>\$[a;b\mathclose{[}]\\$</code>
<code>\$]a;b[\$\</code>	<code>\$\mathopen{]}a;b\mathclose{[}]\\$</code>

Définir des commandes pour les quatre types d'intervalles est indispensable (voir § 9, n° 1, p. 20) ; en plus de permettre de régler les problèmes d'espacement, cela permet aussi de changer facilement la présentation (par exemple en mettant ou pas une espace fine avant le point-virgule ou en remplaçant le point-virgule par une virgule).

8. — *Confusions entre \backslash et \setminus.* La barre contre-oblique a deux variantes, `\backslash` (pour les quotients à gauche) et `\setminus` (pour la différence de deux ensembles). Voici des exemples sur les formules suivantes.

$$G \backslash X \quad \mathbb{R} \setminus \{1\}$$

incorrect	correct
<code>\$G \setminus X\$\</code>	<code>\$G \backslash X\$\</code>
<code>\$\mathbb{R} \backslash \{1\} \$\</code>	<code>\$\mathbb{R} \setminus \{1\} \$\</code>

9. — *Taper correctement les prime, seconde, tierce, etc.* Il faut utiliser uniquement le caractère ' (touche 4 dans le haut du clavier sur un PC, à répéter le bon nombre de fois selon le nombre de symboles voulus : un pour prime, deux pour seconde, trois pour tierce, etc.), pas ' ou " ou ". Voici des exemples sur les formules suivantes.

incorrect	correct
<code>\$f'(x) \$\</code>	<code>\$f'(x) \$\</code>
<code>\$f''(x) \$\</code>	<code>\$f''(x) \$\</code>
<code>\$f^{'}(x) \$\</code>	<code>\$f'(x) \$\</code>

10. — *Confusions entre certaines flèches.* Ne pas confondre \rightarrow et \mapsto , elles n'ont pas la même utilisation. Voici des exemples sur les formules suivantes.

$$f: X \rightarrow Y \quad f: x \mapsto x^2$$

incorrect	correct
<code>\$f \colon X \mapsto Y\$</code>	<code>\$f \colon X \to Y\$</code>
<code>\$f \colon x \to x^2\$</code>	<code>\$f \colon x \mapsto x^2\$</code>

11. — Confusions entre certaines flèches doubles. Pour les implications et les flèches d'équivalence, penser à utiliser `\implies` et `\iff`. Voici des exemples sur les formules suivantes.

$$A \iff B \quad P \implies Q$$

incorrect	correct
<code>\$A \Leftrightarrow B\$</code>	<code>\$A \iff B\$</code>
<code>\$A \Longleftrightarrow B\$</code>	<code>\$A \iff B\$</code>
<code>\$P \Rightarrow Q\$</code>	<code>\$P \implies Q\$</code>
<code>\$P \LongRightarrow Q\$</code>	<code>\$P \implies Q\$</code>

12. — Confusions entre les notations pour le produit. Un produit ne se note pas avec un astérisque `*` (qui s'utilise pour la convolution, par exemple) ni avec un point `.`, mais avec soit rien du tout, soit un point surélevé « `\cdot` », soit une croix « `\times` ». Voici des exemples sur les formules suivantes.

$$a \cdot b \quad 2 \times 3$$

incorrect	correct
<code>\$a.b\$</code>	<code>\$a \cdot b\$</code>
<code>\$2*3\$</code>	<code>\$2 \times 3\$</code>

13. — Confusions entre `\cup` et `\bigcup`. Faire attention, il y a deux symboles pour l'union : `\cup` et `\bigcup`. Ces deux commandes n'ont pas le même rôle. On utilise `\cup` pour faire l'union de deux ensembles (de même qu'on utilise `+` pour ajouter deux nombres), mais on utilise `\bigcup` lorsqu'on fait une union indexée par une famille (de même qu'on utilise \sum pour une somme indexée par une famille). Voici des exemples sur les formules suivantes.

$$A \cup B \quad \bigcup_{i \in I} A_i$$

incorrect	correct
<code>\$A \bigcup B\$</code>	<code>\$A \cup B\$</code>
<code>[\cup_{i \in I} A_i]</code>	<code>[\bigcup_{i \in I} A_i]</code>

La même chose vaut pour `\cap` (intersection), `\wedge`, `\vee`, `\oplus`, `\otimes`, etc. qui ont des variantes `\bigcap`, `\bigwedge`, etc. Voici un petit tableau récapitulatif (non exhaustif).

version binaire		version en famille		version binaire		version en famille	
<code>+</code>	$a + b$	<code>\sum</code>	$\sum_{i \in I} a_i$	<code>\wedge</code>	$a \wedge b$	<code>\bigwedge</code>	$\bigwedge_{i \in I} a_i$
<code>\times</code>	$a \times b$	<code>\prod</code>	$\prod_{i \in I} a_i$	<code>\vee</code>	$a \vee b$	<code>\bigvee</code>	$\bigvee_{i \in I} a_i$
<code>\amalg</code>	$a \amalg b$	<code>\coprod</code>	$\coprod_{i \in I} a_i$	<code>\cap</code>	$A \cap B$	<code>\bigcap</code>	$\bigcap_{i \in I} A_i$
<code>\oplus</code>	$F \oplus G$	<code>\bigoplus</code>	$\bigoplus_{i \in I} F_i$	<code>\cup</code>	$A \cup B$	<code>\bigcup</code>	$\bigcup_{i \in I} A_i$
<code>\otimes</code>	$f \otimes g$	<code>\bigotimes</code>	$\bigotimes_{i \in I} f_i$	<code>\sqcup</code>	$U \sqcup V$	<code>\bigsqcup</code>	$\bigsqcup_{i \in I} U_i$

14. — Confusions entre \wedge , \bigwedge et Λ . En plus de la confusion possible entre \wedge (`\wedge`) et \bigwedge (`\bigwedge`), on peut aussi confondre avec la lettre grecque majuscule Λ (Lambda). Voici des exemples sur les formules suivantes.

$$\Lambda = z\mathbb{Z} + w\mathbb{Z} \quad \mathbb{C}/\Lambda \quad \vec{w} = \vec{u} \wedge \vec{v}$$

incorrect	correct
$\wedge = z\mathbb{Z} + w\mathbb{Z}$	$\Lambda = z\mathbb{Z} + w\mathbb{Z}$
\mathbb{C}/\bigwedge	\mathbb{C}/Λ
$\vec{w} = \vec{u} \Lambda \vec{v}$	$\vec{w} = \vec{u} \wedge \vec{v}$

15. — Taper correctement les petits o et les grand O . Pour faire un petit o (par exemple, $o(x)$) ou un grand O (par exemple, $O(x^2)$), on utilise les lettres o et O , pas des symboles comme \circ (`\circ`) ou \bigcirc (`\bigcirc`). Voici des exemples sur les formules suivantes.

$$O(\ln x) \quad o(\sqrt{x})$$

incorrect	correct
$\bigcirc(\ln x)$	$O(\ln x)$
$\circ(\sqrt{x})$	$o(\sqrt{x})$

§ 9. Bien choisir ses raccourcis pour les maths

Les macros personnelles sont des éléments importants pour assurer la cohérence d'un document (et, éventuellement, le moyen de changer très facilement de notation). Il ne faut pas définir des macros comme des raccourcis pour taper du texte. Par exemple, une macro comme

```
\newcommand{\forallin}[2]{\forall #1 \in #2}
```

est une mauvaise idée : elle brouille la lecture du fichier source sans permettre d'accéder au sens.

Voici quelques exemples de macros courantes qui peuvent être utiles¹⁰.

10. Ces macros ne sont pas forcément toujours optimales, mais constituent un bon point de départ.

1. — Intervalles. Afin de s’assurer que les intervalles ont un espacement correct, on peut utiliser les commandes suivantes ¹¹.

```
\newcommand{\intervalle}[4]{\mathopen{#1}#2
\mathclose{}}\mathpunct{}};#3
\mathclose{#4}}
\newcommand{\intervalleff}[2]{\intervalle{[}{#1}{#2}{]}}
\newcommand{\intervalleof}[2]{\intervalle{]}{#1}{#2}{]}}
\newcommand{\intervallefo}[2]{\intervalle{[}{#1}{#2}{[}}
\newcommand{\intervalleoo}[2]{\intervalle{]}{#1}{#2}{[}}
```

Noter la définition en deux étapes : d’abord une commande pour choisir comment formater les intervalles, puis la définition des quatre types d’intervalles. Ces commandes s’utilisent ainsi.

code	résultat
<code>\intervalleff{-1}{1}\$</code>	$[-1; 1]$
<code>\intervalleof{-1}{1}\$</code>	$] -1; 1]$
<code>\intervallefo{-1}{1}\$</code>	$[-1; 1[$
<code>\intervalleoo{-1}{1}\$</code>	$] -1; 1[$
<code>\chi_{\intervalleoo{-1}{1}}\$</code>	$\chi_{]-1; 1[}$
<code>\intervalle\llbracket 1\rrbracket \$</code>	$\llbracket 1; n \rrbracket$

La dernière ligne nécessite le package `stmaryrd` et c’est probablement une bonne idée de définir un raccourci pour ces intervalles d’entiers :

```
\newcommand{\intervalleentier}{\intervalle\llbracket #1\rrbracket #2}
```

2. — Ensembles de nombres. C’est une bonne idée d’avoir des raccourcis pour les ensembles de nombres \mathbb{N} , \mathbb{Z} , \mathbb{Q} , \mathbb{R} ou \mathbb{C} .

```
\newcommand{\ensemblenombre}[1]{\mathbb{#1}}
\newcommand{\N}{\ensemblenombre{N}}
\newcommand{\Z}{\ensemblenombre{Z}}
\newcommand{\Q}{\ensemblenombre{Q}}
\newcommand{\R}{\ensemblenombre{R}}
\newcommand{\C}{\ensemblenombre{C}}
```

Noter la définition en deux étapes : d’abord une commande pour choisir comment formater les ensembles de nombres puis leur définition elle-même. Ainsi, si on veut mettre les ensembles de nombre en gras, il suffit de changer la macro `\ensemblenombre`, tout le reste sera automatiquement modifié. Voici des exemples d’utilisation :

11. On a mis un `\mathclose{}}\mathpunct{}}` au lieu d’un simple `\`, afin que l’espace disparaisse lorsque l’intervalle est en exposant ou en indice. Ce genre de raffinement n’est pas exigible pour ce module.

code	résultat
<code>\R \setminus \{1\}</code>	$\mathbb{R} \setminus \{1\}$
<code>\forall n \in \mathbb{N}</code>	$\forall n \in \mathbb{N}$
<code>\mathbb{Z}_+ = \mathbb{N}</code>	$\mathbb{Z}_+ = \mathbb{N}$

3. — Éléments différentiels. Voici une commande fonctionnant relativement bien pour taper les éléments différentiels dans des contextes variés.

`\newcommand{\diff}{\mathop{}\!\mathopen{}\mathrm{d}}`

Voici des exemples d'utilisation de cette commande.

code	résultat
<code>\int_a^b f(t) \diff t</code>	$\int_a^b f(t) dt$
<code>\frac{\diff f}{\diff x}</code>	$\frac{df}{dx}$
<code>\diff f = f'(x) \diff x</code>	$df = f'(x) dx$

4. — Valeur absolue, normes. Voici un version simple avec taille automatique des délimiteurs :

`\newcommand{\abs}[1]{\left\lvert#1\right\rvert}`
`\newcommand{\norme}[1]{\left\lVert#1\right\rVert}`

Voici des exemples d'utilisation de ces commandes.

code	résultat
<code>\abs{-x} = \abs{x}</code>	$ -x = x $
<code>\abs{\frac{x}{y}} = \frac{\abs{x}}{\abs{y}}</code>	$\left \frac{x}{y}\right = \frac{ x }{ y }$
<code>\norme{-x} = \norme{x}</code>	$\ -x\ = \ x\ $
<code>\norme{\frac{x}{a}} = \frac{\norme{x}}{\abs{a}}</code>	$\left\ \frac{x}{a}\right\ = \frac{\ x\ }{ a }$

REMARQUE. — Une autre façon de faire est d'utiliser le package `mathtools` et sa commande `\DeclarePairedDelimiter` qui permet, au besoin, de choisir manuellement la taille des délimiteurs. Il faut faire de choix avant de taper le document, car la syntaxe est différente.

5. — Petits o et Grands O. Voici des macros pour les petits o et grands O avec taille automatique des parenthèses¹² :

`\newcommand{\petito}[1]{\mathopen{}\!\left(#1\right)}`
`\newcommand{\grandO}[1]{\mathopen{}\!\left(0\right)}\mathopen{}\!\left(#1\right)`

Voici des exemples d'utilisation de cette commande.

12. Les `\mathopen{}\!` sont là uniquement pour supprimer une espace parasite avant le `\left`.

code	résultat
<code>\ln(1+x) = x + \petito{x^2}</code>	$\ln(1+x) = x + o(x^2)$
<code>\cotan x = \grand0{\frac{1}{x}}</code>	$\cotan x = O\left(\frac{1}{x}\right)$

(`\cotan` se définit par `\DeclareMathOperator{\cotan}{cotan}`)

6. — Ensembles des ... tels que ... Voici des macros pour les ensembles avec taille automatique des accolades :

`\newcommand{\enstq}[2]{\left\{\#1\mathrel{}\middle|\mathrel{}\#2\right\}}`

Voici des exemples d'utilisation de cette commande.

code	résultat
<code>\enstq{x\in X}{f(x)=0}</code>	$\{x \in X \mid f(x) = 0\}$

7. — Produits scalaires. Le produit scalaire a beaucoup de notations possibles. Pour le représenter, on peut définir une commande à 2 arguments.

`\newcommand{\prodscal}[2]{\left\langle\#1,\#2\right\rangle}`

Voici un exemple d'utilisation de cette commande.

code	résultat
<code>\prodscal{\vec{u}}{\vec{v}}</code>	$\langle \vec{u}, \vec{v} \rangle$

Il y a bien sûr d'autres possibilités pour le produit scalaire :

`\newcommand{\prodscal}[2]{\#1\cdot\#2}`
`\newcommand{\prodscal}[2]{\left(\#1\middle|\#2\right)}`

8. — Restriction. Comme mentionné dans le § 8, n° 3, p. 15, pour la restriction, il vaut mieux éviter d'utiliser `|`, car il peut y avoir des problèmes d'espacement dans certains cas. La bonne façon de faire est de définir une commande spéciale pour cela :

`\newcommand{\restreinta}{\mathclose{}\mid\mathopen{}\hspace{-1pt}}`

On utilisera alors `\sin\restreinta_X` au lieu de `\sin|_X` afin d'obtenir $\sin|_X$ au lieu de $\sin|_X$ (qui a une espace parasite en trop avant la barre verticale).