

Ceci est un paragraphe de texte standard, non inclus dans une liste. Il sert principalement aux introductions ou aux explications générales, mais il est honnêtement peu utilisé dans les fiches de synthèse.

Section typique

- Ceci est un item de premier niveau. Un même item peut contenir plusieurs lignes de texte sans difficulté. Pour revenir à la ligne, on utilise deux antislashs.
C'est ainsi qu'on effectue un retour à la ligne dans un même item, afin de compléter le propos, par exemple.
- Ceci est un autre item. Il contient une sous-liste pour détailler certains points :
 - Ceci est un sous-item.
 - Un autre sous-item pour l'exemple.
 - Attention : les sous-items ne doivent être utilisés que lorsqu'il s'agit de distinguer différents cas (et qu'un tableau ne s'y prête pas). Cela signifie que, pour des remarques, ajouts ou successions d'informations, on ne les utilise pas : on se contente simplement de retours à la ligne.
- L'usage d'un registre de langue soutenu me plaît particulièrement, car il favorise la rigueur et la précision dans mon apprentissage. Par ailleurs, la clarté et la concision demeurent des principes essentiels.
- J'utilise ce dernier item pour rappeler qu'il n'est pas permis de faire usage de mises en relief (gras, italique, souligné, etc.) dans les paragraphes et les items.

Usage des sous-sections

Exemple de sous-section

- L'usage de sous-sections permet d'organiser plus finement le contenu lorsque la situation l'exige.
- Cette pratique reste relativement rare, mais elle demeure préférable à l'emploi d'un simple item suivi d'une multitude de sous-items, ce qui diluerait inutilement la hiérarchie du document.

Principes d'utilisation

- Une sous-section doit toujours être accompagnée d'au moins une autre sous-section au sein de la même section ; sans cela, elle perd sa raison d'être.
- Il convient de rappeler qu'une sous-section constitue un espace classique pour accueillir des éléments tels que des items, des images, des tableaux ou des formules.
- En ce qui concerne les conditions d'usage, un ensemble de sous-sections n'est justifié que lorsqu'il existe un parallèle explicite entre plusieurs éléments abordés successivement. Chacun est alors présenté dans une sous-section dédiée. En revanche, dans la majorité des cas, lorsqu'il s'agit simplement d'énumérer des informations (propriétés, formules, notions, etc.), une liste d'items suffit dès lors qu'aucune analogie forte ne lie plusieurs groupes d'items.

- En somme, un ensemble de sous-sections peut souvent s’interpréter comme un tableau conceptuel dont la structure aurait pu être matérialisée sous forme de tableau, mais dont le format serait parfois trop volumineux ou moins lisible que des sous-sections distinctes.
- En vertu des règles énoncées, le présent document n’illustre pas un usage idéal des sous-sections. Ce choix vise à montrer plusieurs sous-sections à la suite, afin de servir d’exemple.

Section avec des formules

- Une formule mathématique peut être insérée et centrée sur sa propre ligne, de cette manière :

$$s(t) = \langle s(t) \rangle + \sum_{n=1}^{\infty} A_n \cos(n\omega t + \varphi_n).$$

On veille toujours à la ponctuation, hautement considérée, même dans les formules.

- Il est possible de présenter plusieurs formules courtes sur une seule ligne pour économiser de l’espace :

$$f = \frac{1}{T}; \quad T = \frac{1}{f}; \quad \omega = 2\pi f.$$

Tableaux et images

- Il est pertinent d’utiliser des tableaux dès que possible : leur mise en forme visuelle contribue à l’intellect.
- Voici un petit tableau centré, avec des en-têtes non gras :

| | | | |
|--------|------|------|------|
| Atome | H | C | N |
| χ | 2,20 | 2,55 | 3,04 |

- Voici un tableau plus courant :

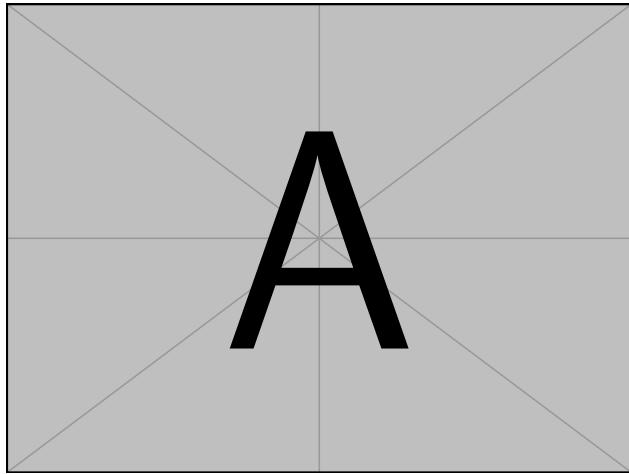
| Catégorie | Formule | Nom |
|------------------|----------------------------------------------------------------------------|-------------------------------------------------|
| Molécule neutre | NH_3 | Ammoniac |
| Cation (positif) | H_3O^+ NH_4^+ | Oxonium Ammonium |
| Anion (négatif) | OH^- SO_4^{2-} NO_3^- MnO_4^- | Hydroxyde Sulfate Nitrate Permanganate |

- Et voici un tableau plus conséquent, occupant toute la largeur du texte :

| Colonne | Autre colonne | Une autre |
|-------------------|-----------------------------------------|---------------------------|
| Exemple | Contenu de la case | Autre contenu |
| C’est un tableau | En effet | Un tableau de qualité |
| Pour quoi faire ? | Idéal lorsqu’il y a beaucoup de contenu | Pour ne pas s’y perdre... |

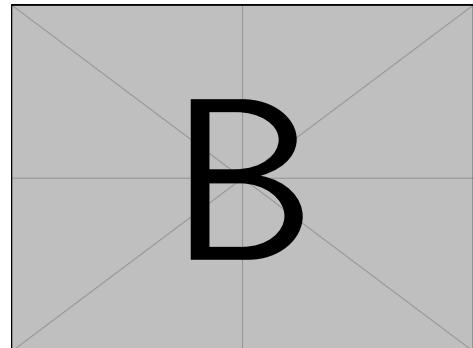
Lorsqu’un tableau contient un contenu conséquent, il est préférable d’y appliquer les règles usuelles de ponctuation.

- Il est également aisément d’insérer une image et de la centrer :



Mise en page avancée

- L'environnement `minipage` permet de créer des blocs de contenu.
- On peut ainsi placer une liste d'éléments à côté d'une illustration, ce qui fait à la fois gagner de la place et permet d'illustrer directement le propos.
- La largeur de chaque bloc est définie en pourcentage. Veillons à ce que leur somme ne dépasse pas 1 :)
- Pour les fiches de type formulaire, l'environnement `multicols` est idéal. Il permet au texte de remplir la colonne de gauche, puis de passer automatiquement en haut de la colonne de droite, économisant ainsi l'espace vertical.
- Ceci est un item dans la première colonne.
- On peut se comporter ici comme dans un environnement classique (titres de section, items, tableaux...).
- Et ainsi de suite, jusqu'à ce que la colonne soit pleine...
- ... ou que l'on décide de passer manuellement à la suivante.
- Nous voici dans la deuxième colonne !
- C'est un outil très pratique pour les listes de définitions ou les formulaires.
- La sublime ligne de séparation est définie dans le préambule.



Exemple de code informatique (Python)

- On peut faire référence à des éléments de code directement dans le texte, par exemple en mentionnant la fonction `calculerValeur()` ou la variable `resultat`.
- Pour présenter un algorithme, on recourt à un bloc de code. Oh, en voici justement un exemple !

```
1 def calculer_aire(rayon): # Calcule l'aire d'un cercle
2     if rayon < 0:
3         return "Rayon non valide."
4     else:
5         aire = 3.14 * (rayon ** 2)
6         return aire
```