

Alignement horizontal

Qu'est-ce qui est le plus joli : $\sqrt{a}\sqrt{b} = \sqrt{ab}$ ou $\sqrt{a}\sqrt{b} = \sqrt{ab}$? $(O; \vec{i}, \vec{j}, \vec{k})$ ou $(O; \vec{i}, \vec{j}, \vec{k})$? Évidemment quand on écrit des mathématiques, ce qui compte en premier, c'est la justesse de ce que l'on écrit ; mais l'intérêt de L^AT_EX c'est aussi d'écrire de belle façon les formules. C'est d'ailleurs pour ça que L^AT_EX a été conçu.

1 `\baselineskip`

La distance verticale entre deux lignes est donnée par la variable `\baselineskip`. Cette variable a une valeur qui dépend de la police dans laquelle on écrit et de sa taille.

Je n'ai pas trouvé comment afficher la valeur de cette variable !

On peut modifier cette variable au moyen du coefficient multiplicateur `\baselinestretch` en redéfinissant ce coefficient qui est à 1 par défaut : `\renewcommand{\baselinestretch}{2}` double l'interligne.

Cette commande doit être immédiatement suivie de `\selectfont` si elle est située dans le corps du texte, c'est-à-dire après `\begin{document}` ; c'est inutile si elle est située dans le préambule.

2 `\rule`

Pour bien comprendre ce qui se passe, il faut détailler un peu l'instruction `\rule`.

Cette instruction crée une boîte rectangulaire dont on doit donner la largeur et la hauteur ; il y a une option pour décaler la boîte par rapport à la ligne de texte : `\rule[position]{largeur}{hauteur}`.

Voyons trois exemples d'utilisation de `\rule` :



Le premier rectangle est positionné sur la ligne, le deuxième est décalé de 0,2 cm au-dessus de la ligne et le troisième de 0,2 cm en-dessous.

On peut tracer une ligne, par exemple d'épaisseur 0,2 point, avec l'instruction `\rule`, même si ce n'est pas sa fonction première :

Il suffit de taper `\rule{\linewidth}{0.2pt}` (`\linewidth` désigne la largeur de la ligne courante).

J'utilise souvent cette instruction en prenant une largeur nulle pour le rectangle ; cela forme ainsi une « béquille » qui permet d'agrandir certaines interlignes pour écrire des fractions, par exemple :

$\frac{13}{7}$ (regardez l'espacement qu'il y a au-dessus du 13).

Enfin, je faisais ça avant d'avoir compris ce qu'étaient les instructions `\strut` et `\mathstrut` !

3 `\strut`

L'instruction `\strut` est une forme particulière de `\rule` ; d'après le Wikibook [Latex.pdf](#), c'est exactement `\rule[-0.3\baselineskip]{0pt}{\baselineskip}`.

L'instruction `\strut` correspond donc à un rectangle de largeur nulle, positionné à 0,3 espacement de ligne en-dessous de la ligne courante, et d'une hauteur totale d'un espacement de ligne. Comme `\strut` n'est qu'une `\rule` particulière, on peut l'utiliser aussi bien en mode texte qu'en mode mathématique.

On va utiliser le mode mathématique pour bien voir l'effet de `\strut` ; voici ce que cela donne en coinçant une lettre entre un soulignement (`\underline`) et un surlignement (`\overline`) :

$$\underline{\overline{aa}} \quad \overline{\underline{bb}}$$

Ce résultat est obtenu en écrivant successivement :

- `$$\underline{\overline{a}}$$`
- `$$\underline{\overline{a \strut}}$$`
- `$$\underline{\overline{a \rule[-0.3\baselineskip]{0pt}{\baselineskip}}}$`

puis en recommençant avec la lettre *b*.

On voit bien ainsi que `\strut` n'est qu'un raccourci pour `\rule[-0.3\baselineskip]{0pt}{\baselineskip}`.

4 `\mathstrut`

Malgré l'abondante documentation que l'on peut trouver sur la toile, je n'ai rien trouvé comme explication convaincante sur `\mathstrut` sinon quelque chose du style : « c'est comme `\strut` mais pour le mode mathématique ». Pour moi, c'est un peu léger comme explication, d'autant que l'on peut utiliser `\strut` en mode mathématique !

On regarde d'abord la différence entre `\strut` et `\mathstrut` :

$$\underline{\underline{aa}}$$

obtenu par `$$\underline{\overline{a \strut}}\underline{\overline{a \mathstrut}}$$`.

On peut donc penser que `\mathstrut` est un `\rule` particulier dont il faut déterminer les paramètres.

Après quelques essais, j'ai trouvé que `\rule[-0.2\baselineskip]{0pt}{0.82\baselineskip}` avait à peu près le même effet que `\mathstrut`.

En reprenant l'exemple du dessus en remplaçant `\strut` par `\mathstrut`, et en ajustant les valeurs dans `\rule`, cela donne :

$$\underline{\underline{aa}} \quad \overline{\overline{bb}}$$

5 Applications

Revenons à ce qui était écrit en début de chronique en définissant certaines commandes.

- **Racine carrée**
 - Nouvelle commande `\rac` : `\newcommand{\rac}[1]{\sqrt{#1\mathstrut}}`
 - Utilisation : quels que soient *a* et *b* positifs, $\sqrt{a \times b} = \sqrt{a} \times \sqrt{b}$.
 - Obtenu par : `$$\rac{a\times b} = \rac{a} \times \rac{b}$$`
- **Conjugué**
 - Nouvelle commande `\barre` : `\newcommand{\barre}[1]{\overline{#1\mathstrut}}` (merci Denis VERGÈS).
 - Utilisation : $\overline{a \times b} = \overline{a} \times \overline{b}$.
 - Obtenu par : `$$\barre{a\times b} = \barre{a} \times \barre{b}$$`

• **Repère**

- Nouvelle commande `\Oijk` :

$$\newcommand{\Oijk}{\left(\mathrm{O}\,,\;\backslash,\;\backslash\overrightarrow{\mathrm{i}\mathstrut}\,,\;\backslash\overrightarrow{\mathrm{j}\mathstrut}\,,\;\backslash\overrightarrow{\mathrm{k}\mathstrut}\right)}$$
- Utilisation : soit le repère $(O; \vec{i}, \vec{j}, \vec{k})$.
- Obtenu par : `\Oijk`

• **Vecteur**

- Nouvelle commande `\vect` :

$$\newcommand{\vect}[1]{\overrightarrow{\,#1\,}\mathstrut}$$
- Utilisation : soient les vecteurs \vec{u} et \vec{v} .
- Obtenu par : `\vect{u}` et `\vect{v}`

• **Vecteur défini par deux points**

L'écriture mathématique veut que l'on écrive les noms des points en grandes capitales et en romain, alors que le mode mathématique met automatiquement les lettres en grandes capitales en italique. C'est la fonction `\mathrm` qui convertit d'italique en romain. On peut donc écrire le vecteur \overrightarrow{AB} par `\vect{\mathrm{AB}}`. On peut aussi définir une nouvelle commande qui fait le travail !

- Nouvelle commande `\vectt` (avec deux fois la lettre t) :

$$\newcommand{\vectt}[1]{\overrightarrow{\,\mathrm{#1}\,}\mathstrut}$$
- Utilisation : soient les vecteurs \overrightarrow{AB} et \overrightarrow{MN} .
- Obtenu par : `\vectt{AB}` et `\vectt{MN}`

On peut noter des espaces fines ajoutées avant la première lettre du vecteur et après la seconde dans la définition de la commande, ce qui a pour effet d'allonger un peu la flèche.

Ainsi grâce au `\mathstrut` présent dans les définitions de `\vect` et `\vectt`, quand on écrit

$$\vec{u} = \overrightarrow{AB}$$

, obtenu par `\[\vect{u} = \vectt{AB}\]`, les flèches sont parfaitement alignées !