

L^AT_EX & les illustrations

ou comment insérer des petits ~~mickeys~~ gnus dans ton texte
On verra aussi comment triturer du texte

Bertrand Masson

Les fiches de Bébert

1^{er} juillet 2012

Le package graphicx

Insérer des figures est très simple. Pour cela il faut charger le package `graphicx`, par la commande `\usepackage{graphicx}` dans l'en-tête de ton fichier source. Comme notre but est de fabriquer des fichiers pdf, nous utiliserons uniquement les formats d'images suivants : `.png`, `.jpg` et `.pdf`.

La commande pour placer une image dont le nom du fichier est : `gnu.png` est donc :

```
\includegraphics{gnu.png}
```

Attention

Comme aucun répertoire n'est indiqué, \LaTeX va chercher l'image uniquement dans le répertoire courant, c'est à dire celui qui contient le fichier `.tex`

Déclarer le chemin du répertoire des images

Pour faciliter la gestion de mes documents, je place mes images dans un répertoire nommé « images », lui même situé dans le répertoire où se trouvent mes fichiers \LaTeX . La commande pour placer une image `gnu.png` est donc :

```
\includegraphics{./images/gnu}
```

Tu peux définir le chemin du répertoire où \LaTeX doit aller chercher les images par : `\graphicspath{{./images/}}`. L'expression `./images/`, signifie le répertoire « images » du répertoire courant. Tu places cette commande dans le préambule de ton fichier source (avant `\begin{document}`).

Si tu as classé tes illustrations dans différents répertoires en fonction de leur nature (schémas, photos, courbes), il te faudra utiliser la commande :

```
\graphicspath{{./dossier1/}{./dossier2/}...{./dossierN/}}
```

Attention

Même si tu ne declares qu'un seul répertoire, il doit être entouré de `{}`, et n'oublie pas la barre de fraction (`/`) finale.

Déclarer les extensions des images

Tu peux t'abstenir de préciser l'extension si tu as mis dans le préambule de ton document la commande :

```
\DeclareGraphicsExtensions{.png,.jpg,.pdf}
```

Dans l'exemple ci-dessus, \LaTeX va d'abord chercher l'existence d'un fichier avec l'extension [.png](#) et s'il ne trouve pas, [.jpg](#) et enfin [.pdf](#).

Tu peux bien évidemment changer l'ordre.

Dans les exemples qui suivent, on va considérer que j'ai déclaré le chemin et les extensions.

Notre première image

On va placer notre première image dans un document \LaTeX . Voici le code :

```
Un logo gnu \includegraphics{gnu} et  
un logo debian \includegraphics{debian}.
```

et voici le résultat :



Nos logos sont un peu grands. Par défaut, `\includegraphics` affiche le dessin dans sa taille réelle, soit 125x50 pixels pour le logo gnu.

Pour opérer des changements, `\includegraphics` accepte des options du type `option=valeur`, séparées par des virgules.

```
\includegraphics[option1=valeur1,option2=valeur2]{gnu}
```

Dimensionner les images

Comme on va manipuler des longueurs, je te conseille, si ce n'est déjà fait, la lecture de la fiche « \LaTeX et les longueurs ».

Tu peux commencer par modifier l'échelle de ton image par l'option `scale`. Des valeurs supérieures à 1 augmentent la taille de l'image (`scale=2` double la taille), des valeurs comprises entre 0 et 1 diminuent la taille (`scale=0.5` divise par 2 les dimensions).

```
Un logo gnu \includegraphics[scale=0.3]{gnu} et  
un logo debian \includegraphics[scale=0.5]{debian}.
```


Un logo gnu  et un logo debian .

Dimensionner les images

Tu peux également donner directement les dimensions souhaitées pour l'image. L'option `width` règle la largeur de l'image et `height` sa hauteur. Si tu ne donnes qu'une seule dimension (largeur ou hauteur), les proportions de l'image seront conservées.

Comme le but ici est d'intégrer notre image dans du texte, j'utilise « em » comme unité. Cette valeur équivaut à la longueur de la lettre m.

```
Un logo gnu \includegraphics[width=2em]{gnu} et  
un logo debian \includegraphics[height=2.5em]{debian}.
```

Un logo gnu GNU et un logo debian  .

Dimensionner les images



Tu n'es pas obligé de conserver les proportions :

```
Un logo gnu \includegraphics[width=6em,height=1em]{gnu} et  
un logo debian \includegraphics[width=0.5em,height=3em]{debian}.
```

Un logo gnu  et un logo debian .

L'option `keepaspectratio` demande un booléen prenant les valeurs `true` et `false`. Avec la valeur `true`, les proportions sont respectées. De plus, les valeurs données à `width` et `height` imposent respectivement la largeur et la hauteur maximale de l'image.



```
Un logo gnu \includegraphics[width=6em,height=1em,keepaspectratio=true]{gnu} et  
un logo debian  
\includegraphics[width=0.5em,height=3em,keepaspectratio=true]{debian}.
```

Un logo gnu  et un logo debian .

Rotation des images

Pour faire tourner l'image, on utilise l'option `angle=nombreEnDegrés` :

Un logo gnu `\includegraphics[width=3em,angle=45]{gnu}` et
un autre logo gnu `\includegraphics[width=3em,angle=-45]{gnu}`.

Un logo gnu  et un autre logo gnu .

Tu peux préciser l'origine de la rotation. Auparavant on va aborder la notion de boîte sous `LATEX`.

Un petit GNU gnu

L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

On a donc des boîtes qui contiennent des lettres.



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

On a donc des boîtes qui contiennent des lettres. Puis des boîtes de mots contenant des boîtes de lettres,



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

On a donc des boîtes qui contiennent des lettres. Puis des boîtes de mots contenant des boîtes de lettres, et enfin des boîtes de phrases contenant des boîtes de mots. Le logo gnu est aussi mis en boîte.



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

On a donc des boîtes qui contiennent des lettres. Puis des boîtes de mots contenant des boîtes de lettres, et enfin des boîtes de phrases contenant des boîtes de mots. Le logo gnu est aussi mis en boîte.

Chaque boîte a une origine.



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

On a donc des boîtes qui contiennent des lettres. Puis des boîtes de mots contenant des boîtes de lettres, et enfin des boîtes de phrases contenant des boîtes de mots. Le logo gnu est aussi mis en boîte.

Chaque boîte a une origine.



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

On a donc des boîtes qui contiennent des lettres. Puis des boîtes de mots contenant des boîtes de lettres, et enfin des boîtes de phrases contenant des boîtes de mots. Le logo gnu est aussi mis en boîte.

Chaque boîte a une origine.



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

On a donc des boîtes qui contiennent des lettres. Puis des boîtes de mots contenant des boîtes de lettres, et enfin des boîtes de phrases contenant des boîtes de mots. Le logo gnu est aussi mis en boîte.

Chaque boîte a une origine.

L^AT_EX place cette origine sur une ligne appelée ligne de base. Tu peux remarquer que cette origine n'est pas toujours au coin en bas à gauche (lettres p et g).



L^AT_EX ne compose pas des mots avec des lettres mais manipule des boîtes qui contiennent des objets. Un peu comme les ouvriers typographes et leurs caractères en plomb.

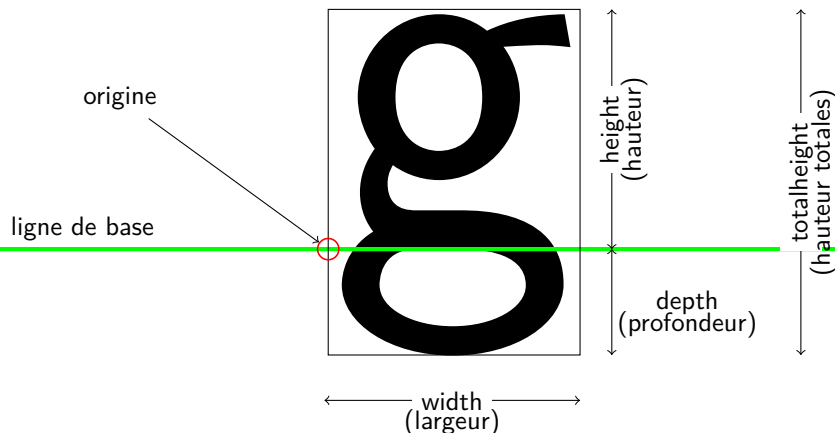
On a donc des boîtes qui contiennent des lettres. Puis des boîtes de mots contenant des boîtes de lettres, et enfin des boîtes de phrases contenant des boîtes de mots. Le logo gnu est aussi mis en boîte.

Chaque boîte a une origine.

L^AT_EX place cette origine sur une ligne appelée ligne de base. Tu peux remarquer que cette origine n'est pas toujours au coin en bas à gauche (lettres p et g).

Les boîtes L^AT_EX ont trois dimensions : une largeur, une hauteur et une profondeur qui correspond à ce qui se trouve sous la ligne de base. Dans notre exemple, toutes les boîtes lettres ont une profondeur nulle à l'exception du p et du g.



En reprenant notre lettre g, voici les différentes longueurs associées à une boîte :





En plus des options `width` et `height` que l'on a vues précédemment, `\includegraphics` reconnaît également `totalheight` et `depth`.

Rotation des images

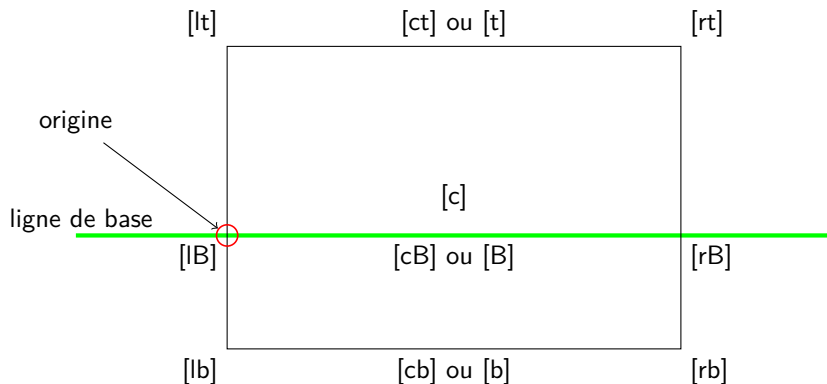
Revenons à notre rotation :

Un logo gnu  et un autre logo gnu .

Comme tu peux le constater, la rotation s'effectue par défaut par rapport à l'origine de la boîte sur la ligne de base. On peut modifier le centre de rotation à l'aide de l'option `origin=valeur`. Les valeurs possibles pour `origin` sont : c (center), b (bottom), t (top), r (right), l (left), B (ligne de base) et des combinaisons de ces valeurs, comme cr ou lt. Tu trouveras un schéma explicatif page suivante. Voici un exemple avec `origin=c` pour le premier gnu et `origin=rt` pour le second.

Un logo gnu  et un autre logo gnu .

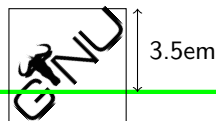
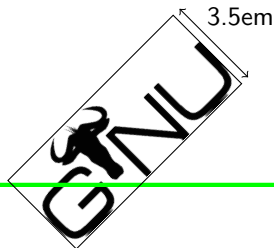
Rotation des images



Rotation des images

Attention l'ordre des options est importante :

```
\includegraphics[height=3.5em,angle=45,origin=c]{gnu}  
\includegraphics[angle=45,height=3.5em,origin=c]{gnu}
```



L^AT_EX exécute les commandes de gauche à droite. Dans le premier cas, il va d'abord définir les dimensions de l'image puis effectuer une rotation. Dans le second cas, il va effectuer une rotation puis appliquer le dimensionnement sur l'objet tourné. Je te rappelle que [height](#) ne concerne que ce qui est au dessus

Inclure des fichiers pdf

L' inclusion d'un fichier pdf se fait de la même façon que n'importe quelle image. Tu as une option supplémentaire `page=`, qui permet de choisir la page importée (par défaut c'est la première). Voici deux exemples :

```
\includegraphics[scale=0.25]{illustration}  
\includegraphics[scale=0.25,page=2,angle=45]{illustration}
```

Les fiches de Bébert

L^AT_EX & les illustrations

ou comment insérer des petits ri-ci-keys dans ton texte
On verra aussi comment insérer du texte

Bertrand Masson

Les fiches de Bébert

1^{er} Février 2010

Le package `graphics`

Insérer des figures est très simple. Pour cela il faut charger le package `graphics` par la commande `\usepackage{graphics}` dans l'en-tête de ton fichier source.
Comme c'est très bas et si tu charges des fichiers pdf, il faut aussi charger le package `graphicx` par la commande `\usepackage{graphicx}` dans l'en-tête de ton fichier source.
Les commandes pour insérer une image `includegraphics` et `includegraphics* (for pdf)`.
Attention : comme la commande `includegraphics` ne fonctionne pas avec les fichiers pdf, il faut utiliser `includegraphics*` pour insérer un fichier pdf. C'est à peu près tout ce qu'il y a à savoir sur ce sujet.

Inclure des fichiers pdf

Il est possible de n'afficher qu'une partie de la page à l'aide des options `trim=a b c d` et `clip`. Les quatre valeurs prises par `trim` correspondent à la quantité d'espace à rogner à gauche pour a, en bas pour b, à droite pour c et en haut pour d. L'unité utilisée est le « inch », (1 in = 72pt ou 2,54 cm). La commande `clip` découpe l'image aux dimensions définies par `trim`. Je trouve que cette option n'est pas très facile à utiliser et je l'utilise de manière empirique, car il n'est pas facile de voir à quoi correspond 1 in sur une page en pdf. Voici un exemple où je centre grossièrement l'image sur le logo gnu de la page 5 du pdf que tu es en train de lire.

```
\includegraphics[scale=0.5,trim=60 60 150 80,clip,page=5]{illustration}
```

résultat :



sont un peu grands. Sans précision \includegraphics : sa taille réelle soit pour le logo gnu
dier à celà \includegraphics accep

Tu peux utiliser cette technique avec n'importe quelle image, mais je trouve plus pratique de préparer convenablement les images avec gimp.

En plus de la commande `\includegraphics`, le package `graphicx` possède d'autres commandes qui permettent de manipuler du texte, ou tout autre objet L^AT_EX :

`\scalebox` et `\resizebox` permettent d'agrandir ou de rétrécir un objet, `\reflectbox`, crée un miroir de l'objet, et `\rotatebox` exécute des rotations.

Manipuler du texte : scalebox

`\scalebox` fonctionne comme l'option `\scale` de `\includegraphics`. La commande agit sur l'échelle d'un objet :

`\scalebox{échelleLargeur}[échelleHauteur]{objetLaTeX}`

Un texte normal, `\scalebox{3}{un très grand texte}` et `\scalebox{0.3}{un tout petit texte.}`

Un texte normal, **un très grand texte** et

un tout petit texte.

`[échelleHauteur]` est entre `[]` et est donc optionnelle. Si tu ne la précises pas, l'objet sera déformé.

`\scalebox{1}[4]{Un texte déformé}`

Un texte déformé

Manipuler du texte : reflectbox

`\reflectbox` et un raccourci pour `\scalebox{-1}[1]{objetLaTeX}` qui permet d'écrire en miroir.

Un texte normal, `\reflectbox{un texte miroir}`

Un texte normal, `\scalebox{-1}[1]{un texte miroir}`

Tu peux t'amuser en variant les facteurs d'échelle :

`\scalebox{1}[-2]{Un texte miroir} \reflectbox{un texte miroir}`

`\scalebox{1}[-2]{Un texte miroir}`
`\reflectbox{un texte miroir}`

L'utilisation d'une `\makebox` de largeur nulle est intéressante. **Attention**, le deuxième argument de la `\makebox` est la lettre L minuscule :

`\makebox[0mm][l]{Un texte miroir}\scalebox{1}[-1]{Un texte miroir}`

`\makebox[0mm][l]{Un texte miroir}`
`\scalebox{1}[-1]{Un texte miroir}`

Manipuler du texte : `resizebox`

Tu peux préciser les dimensions verticale et horizontale avec la commande `\resizebox{dimension horizontale}{dimension verticale}`. Tu peux conserver les proportions en n'indiquant qu'une seule dimension l'autre étant remplacée par un « ! »

```
\resizebox{!}{1cm}{Un texte} et  
\resizebox{1cm}{10mm}{un autre texte}
```

Un texte et un autre texte

Tu peux mettre dans ces boîtes (`\reflectbox`, `\scalebox`, `\resizebox` et `\rotatebox`) n'importe quel élément \LaTeX , tel que listes, autres boîtes, tableaux... Tu trouveras page suivante un exemple avec un tableau.

Manipuler du texte : resizebox

Le premier tableau est normal, le second est créé par la commande suivante :

```
\resizebox{4cm}{3cm}{%  
\begin{tabular}{|c|c|c|}  
\hline  
Banane&Poire&Radis\\\hline  
Choucroute&Andouille&Fraise\\\hline  
Pou&Hibou&Genou\\\hline  
\end{tabular}}
```

Banane	Poire	Radis
Choucroute	Andouille	Fraise
Pou	Hibou	Genou

Banane	Poire	Radis
Choucroute	Andouille	Fraise
Pou	Hibou	Genou

Manipuler du texte : rotatebox

Tu effectues la rotation de texte avec :

`\rotatebox[option=valeur]{angle en degrés}{le texte}`

L'option peut être `origin`, que l'on a vue précédemment et qui prend les mêmes valeurs, et `x=dimension` et `y=dimension`, qui permettent de donner les coordonnées du point de rotation.

Un `\rotatebox{45}{texte}` et un autre `\rotatebox[c]{45}{texte}`
et un troisième `\rotatebox{-45}{texte}`

Un `texte` et un autre `texte` et un troisième `texte`

Un `\rotatebox[x=10mm, y=0mm]{45}{texte}` et un autre
`\rotatebox[x=1mm, y=15mm]{45}{texte}`

Un `texte` et un autre `texte`

On vient de voir les commandes de base pour placer des images et manipuler des objets. Une prochaine fiche montrera comment placer harmonieusement tes figures dans ton texte (la notion de flottants), ajouter des légendes, une table des figures. . .