

# CHAPITRE 2

---

## Style et mise en page

---

Dans ce chapitre, nous allons voir comment mettre en page le texte et modifier le style de ce dernier. On commencera par voir comment modifier l’alignement du texte et insérer des citations ou des poèmes. On verra ensuite comment modifier le style de la police et utiliser les couleurs. Enfin, on terminera avec des mises en page avancées du texte.

### 2.1 Alignement et positionnement

Dans cette section, on va voir trois choses. On va commencer avec l’alignement du texte. Ensuite, on verra comment gérer les espaces et en insérer dans le document. Enfin, on terminera en voyant comment remplir de l’espace avec des traits (continus, pointillés ...).

#### 2.1.1 Alignement


Par défaut, les paragraphes d’un document sont justifiés, c’est-à-dire que le texte est accolé aux marges gauches et droites. Cet effet est obtenu en rendant la largeur des espaces entre les mots variable.

Aligné à gauche

Centré

Aligné à droite

On peut modifier ce comportement et forcer le texte à être collé à gauche ou à droite, ou centrer le texte en utilisant respectivement les commandes `\raggedright`, `\raggedleft` et `\centering`.



**Code**

```
\raggedright Aligné à gauche
\centering Centré
\raggedleft Aligné à droite
```


Il faut bien faire attention que `\raggedright` permet de coller le texte à gauche.

On peut également modifier l'alignement du texte avec les environnements `flushleft`, `flushright` et `center`. La différence est qu'un espace vertical sera ajouté avant le texte de l'environnement étant donné que ces environnements commencent un nouveau paragraphe.

Aligné à gauche

Centré

Aligné à droite



**Code**

```
\begin{flushleft}
  Aligné à gauche
\end{flushleft}
\begin{center}
  Centré
\end{center}
\begin{flushright}
  Aligné à droite
\end{flushright}
```


### 2.1.2 Espacement

Il y a différents types d'espacements possibles. Tout d'abord, on a le choix entre des espacements horizontaux ou verticaux. Ensuite, on peut choisir entre des espaces de longueur fixe ou élastique.

On obtient des espaces horizontaux de longueur fixe avec `\hspace` et des verticaux avec `\vspace`.

Texte précédé de 1.5cm.  
Paragraphe précédé de -1mm.





**Code**

```

Une ligne de texte \\
\hspace{1cm} Pas de décalage \\
\hspace*{1cm} Un décalage de 1 centimètre \\
\hspace*{\fill} Texte poussé à droite


```

La commande `\\` permet d'insérer un retour à la ligne. On peut utiliser l'option de cette commande pour spécifier un certain espacement vertical.

*Voici un texte :*

**HELLO**

Dans l'exemple ci-dessus, on a placé `Hello` suivi d'un retour à la ligne suivi de `/////`. Afin d'avoir les deux textes superposés, on a fait remonter la seconde ligne de `\baselineskip` (la hauteur d'une ligne de texte).



**Code**

```

\it Voici un texte : \\[2mm]
\tt
Hello \\[-\baselineskip]
/////


```

### 2.1.3 Remplissage

On peut remplir des espaces horizontaux de différentes manières. Les commandes `\hrulefill` et `\dotfill` insèrent un espace horizontal élastique rempli avec un trait continu ou des pointillés.

A \_\_\_\_\_ B ..... C

On peut utiliser les différentes commandes sur la même ligne, et elles se partageront l'espace disponible équitablement.



**Code**

```


A \hrulefill B \dotfill C

```

Une autre solution consiste à utiliser la commande `\dashrulefill` du package `nccrules`.

(—— ———)

A ..... B =-=-=-=- C



**Code** nccrules

```
(\dashrule[1mm]{1cm 5mm 1cm}{1}) \\[3mm]
A \dashrulefill{2 4}{1} B \dashrulefill{4 2 1 2}{1 4 1} C
```

La commande `\dashrulefill` prend deux paramètres, chacun de ceux-ci étant une liste de nombres entiers. Le premier nombre représente la longueur du premier trait plein, le second la longueur du premier trait blanc, etc. Le motif est ensuite répété. Le premier paramètre est utilisé pour le motif horizontal et le second pour le motif vertical.

Avec `\dashrulefill{2 4}{1}`, on définit donc pour le motif horizontal un trait plein de largeur 2 suivi d'un blanc de largeur 4. On n'a qu'un seul trait selon l'axe vertical.

Avec `\dashrulefill{4 2 1 2}{1 4 1}`, on définit que le motif horizontal est formé d'un trait plein de largeur 4 suivi d'un blanc de largeur 2, d'un trait plein de largeur 1 et d'un blanc de largeur 2. Verticalement, on a une ligne de hauteur 1, un blanc de hauteur 4 et à nouveau une ligne de hauteur 1.

Contrairement à `\dashrulefill`, la commande `\dashrule` ne répète pas le motif. Les deux commandes prennent également une option qui indique une longueur de laquelle le trait sera remonté verticalement.

## 2.2 Césure

L<sup>A</sup>T<sub>E</sub>X est capable d'automatiquement découper les mots lorsque ceux-ci sont trop longs pour tenir sur la ligne courante et si le fait de faire passer tout le mot sur la ligne suivante est inesthétique. Pour cela, il se base sur un dictionnaire de césure lui permettant de savoir où découper chaque mot.

Néanmoins, il se peut qu'il ne sache pas où découper certains mots et afin de lui indiquer, vous pouvez utiliser la commande `\-` qui indique les points de découpe possibles. Par exemple, on pourrait écrire :

```
Ceci est an\-\ti\-\cons\-\ti\-\tu\-\tion\-\nel
```

Si vous avez un mot qui revient souvent dans vos documents, il n'est pas pratique de devoir indiquer à chaque fois comment le découper. Pour cela, il vous suffit d'utiliser la commande `\hyphenation` dans le préambule, une fois pour toute.

```
\hyphenation{an-ti-cons-ti-tu-tion-nel}
```

## 2.3 Style

Dans cette section, on va voir les commandes de base pour gérer le style global du document ainsi que celui du texte en particulier.

### 2.3.1 Style du document

Le style global du document est modifié avec la commande `\pagestyle`. On peut également modifier le style d'une seule page avec `\thispagestyle`. Les trois valeurs de base possibles sont :

- `empty` laisse l'entête et le pied de page vides ;
- `plain` ajoute le numéro de page au milieu du pied de page ;
- `headings` insère le titre du chapitre courant et le numéro de page dans l'entête.

Le style par défaut est `plain`. Notez également que le style de la première page de chaque chapitre est différent du style des autres pages.

### 2.3.2 Style de la numérotation

On peut modifier le style de la numérotation de deux manières. Celui-ci sera altéré en fonction de la partie du document, définie avec les commandes vues à la section 1.3.4.

L'autre solution consiste à utiliser la commande `\pagenumbering` qui permet de changer le style de la numérotation à partir de la page courante. Le paramètre de ces commandes peut prendre l'une des valeurs suivantes :

- `arabic` nombres arabes
- `roman` ou `Roman` nombres romains minuscules ou majuscules
- `alph` ou `Alph` lettres de l'alphabet minuscules ou majuscules


### 2.3.3 Style de la police

On peut modifier trois éléments de la police du texte : la famille, la forme et la graisse. Il y a trois familles, quatre formes et deux graisses.

Romain, Sans-serif, Machine à écrire  
 Droit, *Italique*, *Incliné*, PETITE MAJUSCULE  
 Moyen, **Gras**

Il y a plusieurs manières d'obtenir ces styles. Il y a tout d'abord les commandes déclaratives qui modifient le style à partir de l'insertion de

la commande, jusqu'à la fin du bloc courant (délimité par l'accolade fermante).



**Code**

```
{\rmfamily Romain,
\sfamily Sans-serif,
\ttfamily Machine à écrire} \

{\upshape Droit,
\itshape Italique,
\slshape Incliné,
\scshape Petite majuscule} \

{\mdseries Moyen,
\bfseries Gras}
```

Ces commandes existent également en version courte : `\rm`, `\sf`, `\tt`, `\it`, `\sl`, `\sc` et `\bf`. La commande déclarative `\normalfont` permet de remettre le style par défaut (romain, droit, graisse moyenne).

On peut également utiliser les commandes qui vont modifier le style de leur paramètre. Il s'agit des commandes `\textsf`, `\texttt`, `\textit`, `\textsl`, `\textsc`, `\textbf` et `\textnormal`. On va par exemple pouvoir écrire :

```
\textit{Texte en italique} et texte normal
```


à comparer avec la version déclarative :

```
{\it Texte en italique} et texte normal
```

Enfin, il est également possible d'utiliser les environnements `sf`, `sffamily`, `tt`, `ttfamily`, `it`, `itshape`, `sl`, `slshape`, `sc`, `scshape`, `bf` et `bfseries`.

Texte en gras sans-serif. **Ceci est en gras**, en normal, *et en italique*.

Comme vous avez pu le voir sur l'exemple, utiliser les environnements ne crée pas un nouveau paragraphe. De plus, vous voyez qu'il est possible de combiner plusieurs effets en sélectionnant un style dans chaque catégorie. Cependant, certaines combinaisons n'existent pas, comme par exemple petites majuscules grasses.

 **Code**

```
{\bfseries\sf Texte en gras sans-serif.}
\begin{bf}
  Ceci est en gras, \rm en normal, \textit{et en italique}.
\end{bf}
```


Enfin, on peut également emphaser un texte en utilisant la commande `\emph`. Par défaut, un texte emphasé est rendu en italique.

### 2.3.4 Texte souligné

Pour souligner un texte, il suffit d'utiliser la commande `\underline`. Néanmoins, il n'est pas possible de souligner un texte qui dépasse une ligne avec cette commande.

- Voici un très long texte qui est tellement long qu'il ne tient pas sur une seule ligne de texte.
- Voici un très long texte qui est tellement long qu'il ne tient pas sur une seule ligne de texte.


Pour pouvoir souligner un texte qui s'étend sur plusieurs lignes, il faut utiliser la commande `\uuline` du package `ulem`.

 **Code** **ulem**

```
\underline{- Voici un très ... une seule ligne de texte.}
\uuline{- Voici un très ... une seule ligne de texte.}
```

L'insertion du package `ulem` va modifier le comportement de la commande `\emph`. Les textes emphasés seront soulignés. Pour remettre en place le comportement par défaut, il suffit d'utiliser la commande `\normalem`.

Emphase *Emphase*

 **Code** **ulem**

```
\emph{Emphase} \normalem \emph{Emphase}
```

D'autres commandes de soulignement sont proposées dans le package `ulem`. On peut barrer du texte, souligner deux fois, en vague ...




<u>Texte souligné</u>	<code>\uline{Texte souligné}</code>
<u><u>Double souligné</u></u>	<code>\uuline{Double souligné}</code>
<del>Texte barré</del>	<code>\sout{Texte barré}</code>
<del>Texte rayé</del>	<code>\xout{Texte rayé}</code>
<u>~Souligné en vagues</u>	<code>\uwave{Souligné en vagues}</code>
<u>-Souligné en tirets</u>	<code>\dashuline{Souligné en tirets}</code>
<u>.Souligné en points</u>	<code>\dotuline{Souligné en points}</code>

Les deux dernières commandes sont disponibles depuis la toute dernière version du package (21/01/2010). On peut également définir ses propres types de soulignements avec `\markoverwith` dont le paramètre définit le type de trait à utiliser.

Souligné style perso

Remarquez qu'on a utilisé un trait défini avec la commande `\dashrule` du package `nccrules`.



**Code**
`ulem, nccrules`

```

\newcommand\myunderline{
  \bgroup\markoverwith{\dashrule[-0.5ex]{5 2 2 2 5}{1}}\ULon}
\myunderline{Souligné style perso}

```

### 2.3.5 Taille du texte

Pour spécifier la taille du texte, il faut utiliser des commandes qui vont vous permettre de changer la taille du texte. La taille de base est fixée grâce à une option de la commande `\documentclass`. On va par exemple écrire :

```
\documentclass[11pt]{report}
```


Les valeurs possibles sont 10pt, 11pt ou 12pt, la valeur par défaut étant 10pt. Les commandes de taille sont :

<code>\tiny</code>	(minuscule)	<code>\large</code>	(grand)
<code>\scriptsize</code>	(très petit)	<code>\Large</code>	(plus grand)
<code>\footnotesize</code>	(assez petit)	<code>\LARGE</code>	(très grand)
<code>\small</code>	(petit)	<code>\huge</code>	(énorme)
<code>\normalsize</code>	(normal)	<code>\Huge</code>	(géant)

Une autre solution pour modifier la taille de la police est de passer par la commande `\fontsize`.

# ABC

La commande prend deux paramètres qui sont la taille du texte en points et l'espace interligne. La valeur du second paramètre doit environ être 20% supérieur au premier.



**Code**

```
\fontsize{30}{36}
\selectfont
ABC
```


La commande `\selectfont` permet de valider le choix de la nouvelle police. De plus,  $\LaTeX$  impose des limites sur la taille maximale que vous pouvez avoir. Pour contourner cette limite, il suffit d'insérer le package `fix-cm`.

## 2.3.6 Couleur

Grâce au package `color`, on va pouvoir modifier la couleur du texte inséré. La commande `\color` est de type déclarative tandis que la commande `\textcolor` prend en paramètre le texte dont il faut changer la couleur.

Noir, rouge, bleu, encore noir et gris.

Les commandes prennent en paramètre la couleur qu'on souhaite. Différentes valeurs sont prédéfinies : `black`, `white`, `gray`, `red`, `green`, `blue`, `cyan`, `magenta` et `yellow`.



**Code**

```
\bf Noir, {\color{red} rouge, \color{blue} bleu},
encore noir et \textcolor{gray}{gris}.
```

**color**

On peut également définir ses propres couleurs en utilisant la commande `\definecolor`. Celle-ci prend trois paramètres : un nom, un modèle de couleur et les valeurs de la couleur (nombre compris entre 0 et 1). Il existe trois modèles différents :

- `gray` (niveau de gris) une valeur
- `rgb` (RGB) trois valeurs pour rouge, vert et bleu
- `cmyk` (CMYK) quatre valeurs pour cyan, magenta, jaune et noir



**Code** color

```

\definecolor{darkgray}{gray}{0.2}
\definecolor{lightpurple}{rgb}{0.7,0,0.7}
\definecolor{darkgreen}{cmyk}{0.7,0,0.7,0.5}

\textcolor{lightpurple}{Texte en mauve clair}

```

Enfin, notez qu'on peut également spécifier une couleur par ses valeurs avec les commandes `\color` et `\textcolor`. Il suffit de préciser le modèle de couleur en option. On peut par exemple écrire :

```
\textcolor[rgb]{0.99,0.69,0.07}{Texte en orange clair}
```

## 2.4 Mise en page avancée

Dans cette section, on va voir quelques commandes permettant de faire des mises en page avancées de texte. On verra comment mettre en page des poèmes, faire des paragraphes avec des formes particulières, du texte sur plusieurs colonnes et on terminera par voir comment présenter des nombres avec des unités et comment insérer des URLs.


### 2.4.1 Citation

Pour insérer des citations ou des extraits de roman, on peut utiliser les environnements `quote` ou `quotation`. Les marges sont plus larges et la largeur occupée par le texte sur la page est réduite.

Il l'observa, occupée à tailler ses rosiers. Elle avait une silhouette magnifique, toute en longueur, mais pourtant, on la devinait sportive et tonique. Il prit le temps de la détailler.

Sa robe rouge, décontractée mais à la coupe parfaite épousait la moindre des formes de son corps de rêve. Le décolleté était simplement ensorcelant. [...]

La différence entre les deux environnements est que `quotation` ajoute un alinéa devant chaque paragraphe.



**Code**

```

\begin{quote}\small
Il l'observa, occupée à tailler ses rosiers. Elle avait ...
\end{quote}

```

### 2.4.2 Poème

Pour insérer un poème, on peut utiliser l'environnement `verse`. Chacune des lignes du poème doit se terminer par `\\`.

Demain, dès l'aube, à l'heure où blanchit la campagne,  
Je partirai. Vois-tu, je sais que tu m'attends.  
J'irai par la forêt, j'irai par la montagne.  
Je ne puis demeurer loin de toi plus longtemps.



#### Code

```
\begin{verse}\small
Demain, dès l'aube, à l'heure où blanchit la campagne, \\
Je partirai. Vois-tu, je sais que tu m'attends. \\
J'irai par la forêt, j'irai par la montagne. \\
Je ne puis demeurer loin de toi plus longtemps.
\end{verse}
```

### 2.4.3 Forme des paragraphes

On peut obtenir des paragraphes qui ont des formes particulières en utilisant le package `shapepar`. Celui-ci propose des formes prédéfinies, mais il permet également d'en définir des nouvelles.

Une seule  
chose est certaine. Une  
fois que vous aurez goûté  
à  $\LaTeX$ , vous ne pourrez  
plus vous en passer.  
Vous en serez litté-  
ralement amou-  
reux!  
♡



#### Code

**shapepar**

```
\heartpar{\small Une seule chose est certaine. Une fois que vous
aurez goûté à \LaTeX, vous ne pourrez plus vous en
passer. Vous en serez littéralement amoureux !}
```

Les autres formes possibles prédéfinies dans le package sont :

- `\squarepar` : carré (□)
- `\circlepar` : cercle (○)

- `\diamondpar` : diamant (◇)
- `\heartpar` : cœur (♡)
- `\starpar` : étoile à cinq branches (★)
- `\hexagonpar` : hexagone (⬡)
- `\nutpar` : écrou hexagonal

#### 2.4.4 Texte sur plusieurs colonnes

Pour avoir un texte sur plusieurs colonnes, il suffit d'utiliser l'environnement `multicols` disponible avec le package `multicol`. L'environnement prend un paramètre qui est le nombre de colonnes désirées. En option, on peut préciser un texte qui s'étendra sur toutes les colonnes.

##### Chapeau du texte en colonnes

Ce texte s'étend sur plusieurs colonnes qui ont été obtenues grâce à l'environne-	ment <code>multicols</code> disponible en important le package <code>multicol</code> . Un filet de séparation	est obtenu en définissant la longueur <code>columnseprule</code> .
---	---	--

On peut définir l'espacement entre deux colonnes avec la longueur `\columnsep` et la largeur du trait de séparation avec `\columnseprule`.

**Code**
multicol

```

\setlength{\columnsep}{7mm}
\setlength{\columnseprule}{1pt}

\begin{multicols}{3}[\textbf{Chapeau du texte en colonnes}]
\small Ce texte s'étend sur plusieurs colonnes ...
\end{multicols}

```

Pour forcer un saut de colonne, il suffit d'utiliser la commande `\columnbreak`.

#### 2.4.5 Nombre et quantité

Pour mettre en forme proprement des nombres, on peut utiliser la commande `\numprint` du package `numprint`.

```

\numprint{150000.7829}      150 000,782 9
\numprint{2e-15}           2 · 10-15

```

La commande `\numprint` prend une option qui est une unité de mesure. Le nombre et l'unité seront placés dans une seule boîte inséparable. Enfin, on peut également utiliser les commandes `\cntprint` et `\lenprint` pour afficher respectivement des compteurs et des longueurs.

12,5 kg de pommes et 1,5 de lait  
 1 pt = 0,351 46 mm = 0,013 84 in

Remarquez que pour l’affichage d’une longueur, l’option permet de définir dans quelle unité on souhaite afficher la longueur (la valeur par défaut étant le point). Par contre, pour `\numprint`, l’option indique juste quelle unité on souhaite afficher.

**Code**
numprint

```

\ numprint [kg]{12.5} de pommes et \ numprint {1.5} de lait

\ newlength {\one}
\ setlength {\one}{1cm}

\ lenprint {\one} = \ lenprint [mm]{\one} = \ lenprint [in]{\one}

```

### Système International d’unités (SI)

Le package `siunitx` est plus récent et offre des possibilités plus riches. Il permet de représenter les données selon les normes du système international d’unités (SI). Les commandes essentielles sont `\num`, `\si`, `\ang`, `\SI`, `\numrange` et `\SIrange`.

La commande `\num` permet de représenter les nombres.

10 000.123 45	<code>\num{10000.12345}</code>
$1.2 \times 10^{-15}$	<code>\num{1.2e-15}</code>
$5 \times 2 \times 10^{10}$	<code>\num{5 x 2e10}</code>

Les unités sont obtenues avec la commande `\si`. On peut les écrire soit directement, soit en utilisant des commandes. On peut changer le style avec l’option `per-mode`.

$\text{kg m/s}^2$	<code>\si{kg.m/s^2}</code>
$\text{kg m s}^{-2}$	<code>\si{\kilogram\meter\per\second\squared}</code>
$\text{kg m/s}^2$	<code>\si[per-mode=symbol]{\kilogram\meter\per\second\squared}</code>

On formate les angles en utilisant la commande `\ang`. On peut également préciser les minutes et secondes.

$12.7^\circ$	<code>\ang{12.7}</code>
$1^\circ 2' 3''$	<code>\ang{1;2;3}</code>
$1' 10''$	<code>\ang{;1;10}</code>

Enfin, pour insérer un nombre avec une unité associée, on peut soit utiliser `\num` suivi de `\si`, soit utiliser la commande `\SI` qui garantit que le

nombre de l'unité forment un bloc indissociable. L'option de la commande permet de préciser une unité qui vient avant le nombre.


```
15.5 N/m2      \SI{15.5}{N/m^2}
$5.99/kg       \SI[per-mode=symbol]{5.99}[\${\per\kilogram}]
```

Enfin, on peut représenter des intervalles avec les commandes `\numrange` et `\SIRange`.

```
11 to 42          \numrange{11}{42}
120 cm to 180 cm \SIRange{120}{180}{\centi\meter}
```

Grâce à ce package, on pourra par exemple écrire :

Si vous achetez 200 g de pommes à \$4.99/kg,  
vous devrez payer un total de  $\approx 1$  €.



**Code**
sunitx, eurosym

```

Si vous achetez \SI{200}{g} de pommes à \SI[per-mode=symbol]
{4.99}[\${\per\kilo\gram}, \\
vous devrez payer un total de \SI{1}[\$approx$\]{\euro}.
```

### 2.4.6 URL

On peut insérer des URL en utilisant le package `url`. Celui-ci définit une commande `\url` qui permet d'insérer une URL qui pourra être coupée et s'étendre sur plusieurs lignes. On peut définir le style des URL en utilisant la commande `\urlstyle`.

Site web officiel : `\url{http://www.latex-howto.be}`  
Donald Knuth : `\url{http://www-cs-faculty.stanford.edu/~uno/}`

Notez que si l'URL contient les caractères `%` ou `#`, la commande `\url` ne peut pas être utilisée dans le paramètre d'une autre commande.



**Code**
url

```

\urlstyle{sf}

Site web officiel : \url{http://www.latex-howto.be} \\
Donald Knuth : \url{http://www-cs-faculty.stanford.edu/~uno/}
```

Pour pouvoir utiliser une URL avec % ou # dans le paramètre d'une autre commande, il va falloir définir une nouvelle URL avec `\urldef`. Par exemple :

```
\urldef{\mywebsite}\url{http://dom.ext/d%20}
```

Ensuite, on peut utiliser la commande `\mywebsite` n'importe où. On pourra par exemple écrire :

```
\section{Mon site web : \mywebsite}
```

### 2.4.7 Liens hypertextes

Vous pouvez ajouter des liens hypertextes dans un document en utilisant le package `hyperref`. En ajoutant simplement ce package, vous verrez que toute une série de liens seront automatiquement créés pour les entrées de la table des matières, pour les références croisées, pour les notes de bas de page, les URL ...

Il est également possible d'insérer des liens explicitement avec la commande `\href`. Celle-ci prend deux paramètres que sont l'URL du lien et le texte qui doit apparaître dans le texte.

Vous pouvez faire des recherches avec Google par exemple.



**Code** **hyperref**

Vous pouvez faire des recherches avec  
`\href{http://www.google.be}{Google} par exemple.`



## 2.5 Dimensions et format de la page

Dans cette section, on va voir comment modifier le format de la page, ainsi que les dimensions qui définissent la position du texte, des entêtes et pieds de page . . .

### 2.5.1 Maquette

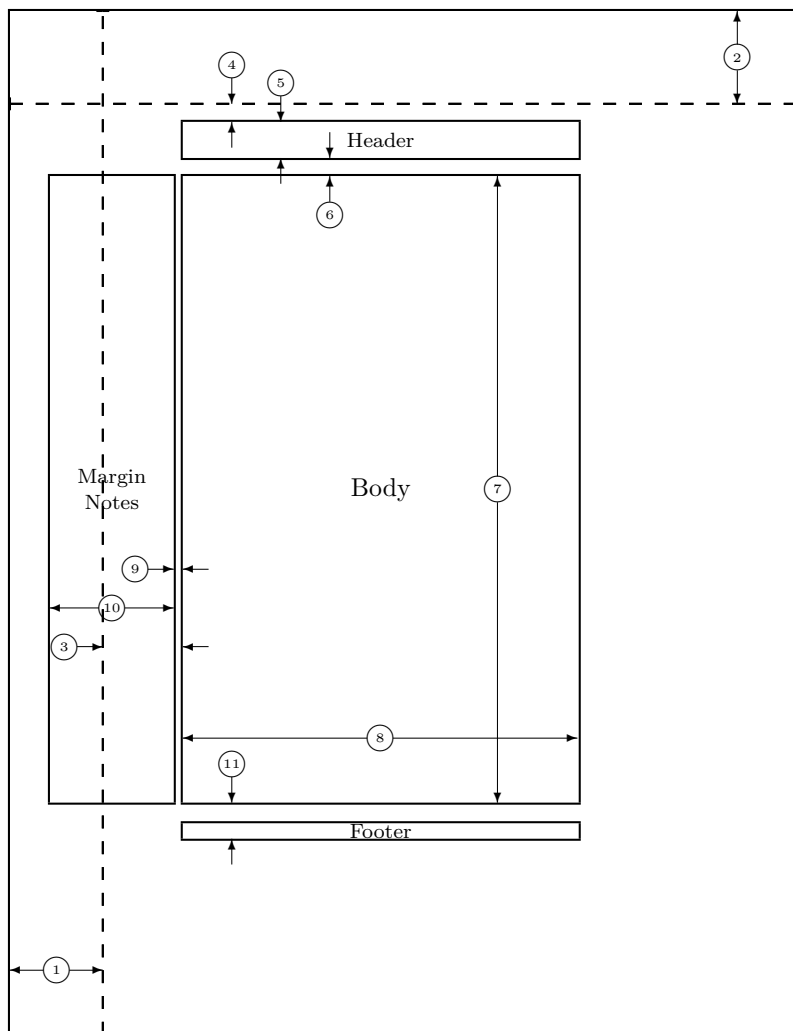
En utilisant la commande `\layout` du package de même nom, on va pouvoir insérer la maquette du document. Il s'agit d'une visualisation d'une page et des différentes dimensions attenantes. On y voit les quatre zones que sont le corps, l'entête, le pied de page et la zone des notes de marge. Toutes les dimensions et espacements sont présentés sur ce dessin. La maquette de ce livre est présentée ci-contre.

Voici les principales longueurs qui définissent un document :

- `\oddsidemargin` : marge sur le côté des pages impaires
- `\topmargin` : marge en haut de page, avant l'entête
- `\headheight` : hauteur de l'entête
- `\headsep` : distance entre l'entête et le haut du corps
- `\textheight` : hauteur du corps du document
- `\textwidth` : largeur du corps du document
- `\marginparsep` : distance entre le corps et les notes de marge
- `\marginparwidth` : largeur des notes de marge
- `\footskip` : distance entre le corps et le pied de page

Enfin, les deux longueurs `\paperwidth` et `\paperheight` contiennent la largeur et la hauteur du papier.

Toutes ces longueurs peuvent être modifiées avec la commande `\setlength`, afin de modifier la mise en page du document. L'utilisation de la maquette s'avère utile pour voir d'un coup d'œil la disposition des éléments sur la page. On verra qu'il y a des packages qui vont rendre ce travail plus facile.



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 62pt	4	\topmargin = 14pt
5	\headheight = 28pt	6	\headsep = 14pt
7	\textheight = 486pt	8	\textwidth = 307pt
9	\marginparsep = 7pt	10	\marginparwidth = 96pt
11	\footskip = 28pt		\marginparpush = 0pt (not shown)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 614pt		\paperheight = 794pt

### 2.5.2 Format prédéfini


En insérant le package `fullpage`, les marges vont être fixées de manière à ce que le texte occupe un maximum de place sur la page. Lors de l'import du package, il faut lui préciser en option le style du document (comme défini par la commande `\pagestyle`, voir section 2.3.1).

On peut également utiliser le package `savetrees` qui, comme son nom l'indique, permet d'économiser le papier. En plus de réduire au maximum les marges, il va également réduire les espacements, la taille de la police, utiliser des abréviations ...

### 2.5.3 Taille du papier

Voyons tout d'abord comment modifier la taille du papier. Par défaut, le papier est au format US letter (un peu plus petit que A4). Pour avoir du papier A4, il suffit de préciser l'option `a4paper` à la commande `\documentclass`.

Une autre manière de préciser le format du papier consiste à utiliser la commande `\setpapersize` du package `vmargin`. En option, on précise l'orientation de la page et en paramètre le format du papier : A0, A1 ..., A9, B0 ..., B9, C0 ..., C9, USletter, USlegal et USexecutive. On peut également donner trois paramètres pour définir une taille personnalisée.


**Code**
**vmargin**

```


\setpapersize{A4} % papier A4
\setpapersize[landscape]{A5} % papier A5, en mode paysage
\setpapersize{custom}{10cm}{2cm} % 10cm de large sur 2cm de haut

```

On peut également utiliser le package `geometry` décrit à la section suivante.

### 2.5.4 Définition des marges

Pour facilement définir les marges d'un document, on peut utiliser la commande `\setmargins` du package `vmargin`. Cette commande prend huit paramètres qui sont respectivement la marge gauche, la marge du haut, la largeur et hauteur du corps, la hauteur de l'entête, l'écart entre l'entête et le corps, la hauteur du pied de page et l'écart entre le corps et le pied de page.

 **Code** vmargin

```
\setmargins{1cm}{1cm}{17cm}{25cm}{1cm}{5mm}{1cm}{5mm}
```

Des variantes de la commande existent, vous les retrouverez dans la documentation du package.

On peut également définir les marges avec le package `geometry`. Les valeurs des différentes longueurs sont spécifiées en option lors du chargement du package. Il y a tout d'abord le corps du document avec son entête, son pied de page et la zone pour les notes de marge. Viennent ensuite les marges gauche, droite, d'en haut et d'en bas. Voici les options les plus courantes :

- `paper` : spécifie la taille du papier (nom)
- `paperwidth` et `paperheight` : spécifient la taille du papier (longueur)
- `width` et `height` : dimension totale du corps du document
- `textwidth` et `textheight` : dimension de la zone du texte
- `includehead`, `includefoot`, `includeheadfoot` et `includemp` : sont utilisés pour indiquer si on souhaite ou non l'entête ou le pied de page ou la zone pour les notes de marge
- `headheight` et `headsep` : spécifient la hauteur de l'entête et la distance entre celle-ci et le texte
- `foot` : spécifie la distance entre la fin du texte et le pied de page
- `left`, `right`, `top` et `bottom` : spécifient les marges gauche, droite, d'en haut et d'en bas
- `hcentering`, `vcentering` et `centering` : permettent de centrer le contenu horizontalement et/ou verticalement

L'exemple donné en haut de page peut donc être obtenu en important simplement le package suivant :

 **Code** geometry

```
\usepackage[paper=a4paper,includeheadfoot,Lleft=1cm,top=1cm,
textwidth=17cm,textheight=25cm,headheight=1cm,headsep=5mm,
foot=5mm]
```

### 2.5.5 Composer un PDF

Le package `pdfpages` permet d'effectuer facilement des opérations sur des fichiers PDF. La première chose qu'on peut faire consiste à insérer certaines pages d'un document PDF dans un document  $\LaTeX$ . On utilise pour cela la commande `\includepdf`. On spécifie les pages que l'on souhaite insérer avec la possibilité de définir des intervalles ou d'insérer une page blanche avec `{}`, avec l'option `pages`.

L'exemple suivant insère la page 1, suivie d'une page blanche, suivie des pages 5 à 9, suivies de la page 15 du document `monDocument.pdf`.

```
\includepdf[pages={1, {}, 5-9, 15}]{monDocument.pdf}
```

Il est également possible d'obtenir plusieurs pages par feuille. On utilise pour cela l'option `nup`. On définit ensuite l'espacement à mettre entre les pages logique avec l'option `delta` et on peut avoir une bordure autour des pages logiques avec l'option `frame`.

Par exemple, pour insérer toutes les pages du document `monDocument.pdf`, avec  $3 \times 2$  pages par feuille, séparées par 5mm et une bordure, il faut écrire :

```
\includepdf[pages=-,nup=3x2,frame]{monDocument.pdf}
```

Enfin, on peut également insérer des pages en provenance de plusieurs fichiers PDF en utilisant la commande `\includepdfmerge` qui possède les mêmes options que `\includepdf` sauf l'option `pages`.

La commande suivante insère la première page du document `docA.pdf`, les pages 5 à 10 du document `docB.pdf` et enfin les pages 3 et 9 du document `docC.pdf`.

```
\includepdfmerge{docA.pdf, 1, docB.pdf, 5-10, docC.pdf, 3, 9}
```

