Titre explicite du rapport d'OCI

Nom des auteurs, classe

26 mars 2023

1. But

Placez ici le but du rapport.

Par exemple, pour un travail sur un site internet, on pourrait écrire :

Notre objectif était de créer un site sur les écureuils avec différentes sections comme un historique, la biologie des écureuils, ...

Le but du présent modèle est de permettre une rédaction facilité de rapports en informatique. Le cœur du modèle tourne donc autour de la possibilité d'écrire du code et de le commenter. Pour cela, le package utilisé est `listings`. Ce module de LaTeX est très complet et relativement complexe à utiliser. Sa documentation (en anglais) se trouve dans le répertoire `documentation` du projet sous le nom `Listings.pdf`. Reportez-vous y fréquemment.

Pour apprendre à l'utiliser, il faut savoir qu'outre l'appel au module et la configuration préalable réalisée dans le préambule de ce document via `\lstset{...}`, la présentation des codes peut se faire en ligne ou hors ligne.

En ligne, la commande est `\lstinline|...|` et le placement du code se fait à la place des points de suspension.

Hors ligne, on utilise l'environnement du listing 1.

Finalement, on peut importer du code d'un fichier externe, comme vous pourrez le voir plus loin. Cela peut être utile pour différentes raisons.

Listing 1 – Placement d'un code en ligne

```
\begin{lstlisting}
  [float,language=...,
  caption={...},
  label={listing:...}]
...
\end{lstlisting}
```

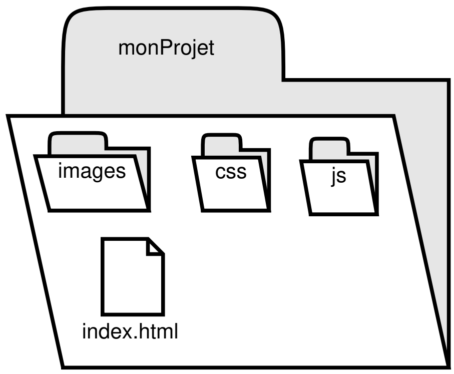
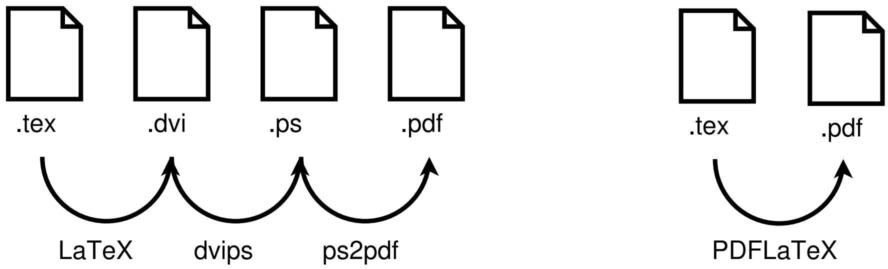


Figure 1 – La structure de fichier d'un site



(a) Compilation via Postscript
Recommandée mais eps

(b) Compilation directe
Moins sure mais jpg et png

Figure 2 – Figures côte à côte

2. Réalisation

On place ici différents éléments pour présenter comment le travail fut réalisé.

Par exemple, toujours pour un site internet, on peut présenter la structure des fichiers comme présenté à la figure 1.

Pour ne pas rendre le modèle trop complexe, le placement des figures côte à côte n'est pas abordé ici. Cependant, s'il vous est nécessaire, consultez le modèle de rapport de travail pratique de physique que j'ai aussi mis à disposition sur le même dépôt de Framagit. Y est détaillé ce type de placement d'images. Vous pouvez aussi consulter le code de la figure 2 qui est reproduit dans le listing 2, qui présente la manière de mettre deux figures côte à côte (en changeant les mots figure et subfigure respectivement par table et subtable, on peut aussi gérer le placement de tableaux côte à côte).

Sur la figure 2, on peut voir les deux manières principales de compiler un document LaTeX. Sur la figure 2a, on voit comment compiler en passant par du Postscript, compilation très puissante et sûre. Elle oblige cependant l'utilisation d'images vectorielles eps exclusivement. Sur la figure 2b, on voit une autre manière de compiler directement en PDF. Elle est moins sûre, mais nécessite ou permet l'utilisation d'images bitmap, jpg et png, à l'exclusion de l'eps.

Listing 2 – Placer des sous-figures

```
\begin{figure}
 \centering
 \begin{subfigure}[b]{0.5\textwidth}
  \centering
  \includegraphics[scale=0.4]{images/....eps}
  \caption[...]{\centering ...}\label{figure:...}
 \end{subfigure}
 \begin{subfigure}[b]{0.4\textwidth}
  \centering
  \includegraphics[scale=0.4]{images/....eps}
  \caption[...]{\centering ...}\label{figure:...}
 \end{subfigure}
 \caption[subfigures]{\centering Figures côte à côte}\label{
    figure:subfigures}
\end{figure}
```

Listing 3 – Un code utilisant turtle

```
from turtle import *  # import du module turtle
forward(100)    # avance de 100
mainloop()    # boucle en attente
```

3. Problèmes rencontrés

Placez ici la description des problèmes rencontrés. Par exemple, en python, si on oublie les parenthèses de la commande `mainloop()`, alors on a une erreur dans un code comme celui du listing 3.

Pour mettre un code à l'intérieur d'une ligne de texte, il faut donc utiliser le code particulier `\lstinline|mainloop()|`. Pour mettre un code en évidence hors ligne, il faut donc par exemple utiliser le code du listing 4.

Mieux, on peut utiliser un fichier extérieur et l'appeler comme présenté dans le lis-

Listing 4 – Placement d'un code hors ligne

```
\begin{lstlisting}
  [float,language=python,
  caption={Un code utilisant turtle},
  label={listing:turtle}]
from turtle import *  # import du module turtle
forward(100)    # avance de 100
mainloop()    # boucle en attente
\end{lstlisting}
```

Listing 5 – Code dans un fichier

```
1 \lstinputlisting[float,
2 language=python,caption={Placement d'un code en ligne},
3 label={listing:horsligne},numbers=left,
4 numberstyle=\tiny,numbersep=6pt]
5 {moncode1.tex}
```

Listing 6 – Un code utilisant html5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Test Javascript</title>
  </head>
  <body>
    <h1>Test Javascript</h1>
    Un langage intéressant.
  </body>
</html>
```

ting 5, ligne 5. On y voit aussi comment activer la numérotation de lignes.

Pour faire référence au numéro d'une ligne présente dans un listing, il faut ajouter à la fin de la ligne le code `(*@\label{listing:nomfichier}@*)`. Les caractères d'échappement parenthèse, étoile et arobase permettant d'interpréter le code LaTeX de référencement `\label{listing:nomfichier}` qui constitue la référence à la ligne sont définis dans le préambule par `\lstset{escapeinside={(*@}{@*)}}`.

4. Analyse

Évidement, on peut aussi spécifier le langage utilisé pour mettre en évidence certains mots clés. Ainsi, en html, on peut écrire le code du listing 6 qui montre la « coloration syntaxique ».

Par ailleurs, on peut régler la numérotation finement. Par exemple, avec le code CSS du listing 7 où on a commencé la numérotation à vingt et numéroté une ligne sur deux.

Beaucoup d'autre options sont disponibles. Reportez-vous à la documentation disponible dans le guide de l'utilisateur mentionné plus haut.

5. Conclusion

Voici un exemple de conclusion. À éviter : mettre ici les remerciements.

Listing 7 – Un code CSS

```css
h1 {
  color: blue;
}
.rouge {
  color: red;
}
```

Ce modèle est entièrement basé sur le module LaTeX *listings*. Plusieurs reproches sont à adresser à ce module (positivement, car il fait déjà beaucoup de choses très bien), dont notamment, de ne pas supporter des langages comme javascript ou css. Évidemment, cela tombe mal pour un mod On peut certes le configurer pour les prendre en charge de manière plus ou moins efficaces, mais il n'est pas déplacé de dire qu'ils pourraient être intégrés à listings.

Ainsi, d'autres modules de traitement des listings peuvent être utilisés, comme *Pygments* ou *LGrind*. Mais, ils ont tous des défauts et le plus complet est certainement listings. Vous trouverez à l'adresse `http://www.grappa.univ-lille3.fr/FAQ-LaTeX/29.53.html` un petit résumé de la situation.

Listing 8 – Un code CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Canvas</title>
    <meta charset="UTF-8" />
    <style>
      #square {
        border: 1px solid black;
        transform: scale(10) rotate(3deg) translateX(0px);
        -moz-transform: scale(10) rotate(3deg) translateX(0px);
      }
      .box {
        transition-duration: 2s;
        transition-property: transform;
        transition-timing-function: linear;
      }
    </style>
  </head>
  <body>
    <canvas id="square" width="200" height="200"></canvas>
  </body>
</html>
```

A. Première annexe

Mettez en annexe tout ce que vous voulez conserver dans le cadre de ce travail pratique, mais qu'il vous semble inutile de lire. On peut penser pour le travail sur le pendule simple à une présentation de l'utilisation de la balance de précision, qui peut s'avérer utile par la suite, mais n'est pas nécessaire pour comprendre votre travail.

Ici, par exemple, dans le listing 8, on peut voir l'utilisation du langage html5 ou dans le listing 9, celle de JavaScript. La coloration syntaxique est partielle, mais si nécessaire, il est possible de l'améliorer dans le fichier `OCI.sty`.

Notez qu'une liste des langage prédéfinis dans listings se trouve dans sa documentation. Mais, autant HTML5, CSS et JavaScript n'y figurent pas. Ils ont été défini dans ce modèle directement et se trouvent dans le fichier `OCI.sty`. Vous pouvez y jeter un coup d'œil, tout en comprenant bien que toute modification de celui-ci peut avoir d'importantes répercutions sur le modèle. Donc prudence.

B. Seconde annexe

Utilisons cette seconde annexe pour présenter divers éléments formels utilisable avec LaTeX dans votre rapport.

Listing 9 – Un code JavaSrcipt

```
<script>
    var canvas = document.createElement('canvas');
    canvas.width = 200;
    canvas.height = 200;
    var image = new Image();
    image.src = 'images/card.png';
    image.width = 114;
    image.height = 158;
    image.onload = window.setInterval(function() {
        rotation();
    }, 1000/60);
</script>
```

Listing 10 – La biliographie

```
1 \begin{thebibliography}{9}
2 \bibitem{EinsteinPR1935}A.~Einstein and N.~Rosen, Phys. Rev. \
    textbf{48}, 73 (1935)
3 \end{thebibliography}
```

B.1. Notes

Les notes de bas de page peuvent être utilisées de la manières suivante[1].

B.2. Références

Dans un document tel qu'un travail de maturité, une bibliographie est indispensable. Si vous désirez en faire une pour votre rapport de physique, il faut tout d'abord simplement mettre une référence d'entrée de bibliographie telle que [1] et mettre la description de l'ouvrage dans l'environnement `thebibliography`, comme présenté dans le listing 10 de ce document. Si vous avez moins de 10 références mettez à la ligne 1 en paramètre le chiffre 9 et 99 si vous en avez moins de cent. Il s'agit là d'une bibliographie très simple obéissant aux canons de la physique. Pour générer des bibliographies plus complexes, utilisez le programme externe *bibtex*.

B.3. Placement des flottants

Techniquement, le placement des éléments graphiques flottants[2] se fait par l'intermédiaire des environnements `figure`, `table`, `subfigure` et `subtable`, pour autant que le

1. Regardez dans le code du modèle.
2. Voir l'excellent site :
`https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions`

module `subcaption` soit chargé.

Ce placement peut être influencé par les éléments :

h pour placer l'élément ici, approximativement,

t pour placer l'élément en haut de page,

b pour placer l'élément en bas de page,

p pour placer l'élément sur une page de flottants,

! pour tenter de forcer LaTeX à faire ce qu'on veut et

H pour placer l'élément exactement ici (nécessite le paquel `float`).

placés entre crochets de la manière suivante :

```
\begin{figure}[tbh...]
```

Références

[1] A. Einstein and N. Rosen, Phys. Rev. **48**, 73 (1935)