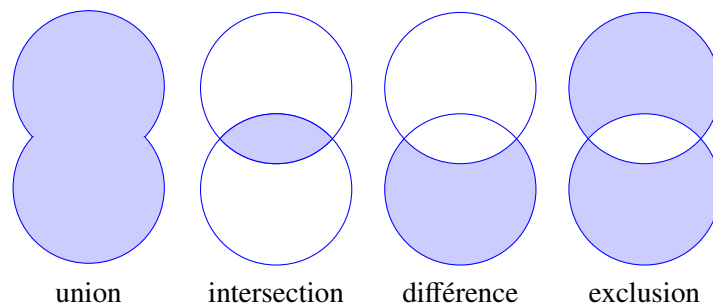


Figures mathématiques avec TikZ (suite)

Avant de continuer, s'assurer d'avoir fini de faire les exercices de la séance n° 5.

9.1 Dessiner des ensembles

Le but de ce § 9.1 est de réaliser les dessins suivants d'union, d'intersection, de différence et d'exclusion de deux ensembles A et B .



On commence par le dessin de la différence, qui est le plus simple à faire. Une commande qui peut être utile pour dessiner séparément le fond du bord du cercle est la commande `\fill`, qui s'utilise comme `\draw`, mais ne dessine pas le bord de l'objet. Par exemple,

```
\begin{tikzpicture}
\fill[fill=green!20] (0,0) rectangle (1,1);
\end{tikzpicture}
```

à comparer à

```
\begin{tikzpicture}
\draw[fill=green!20] (0,0) rectangle (1,1);
\end{tikzpicture}
```

Exercice 1. — En définissant les cercles par des commandes

```
\newcommand{\cercleA}{(0,0.67) circle (1cm)}
\newcommand{\cercleB}{(0,-0.67) circle (1cm)}
```

représenter le cercle B privé de A . On pourra tracer le cercle A en blanc par dessus le cercle B .

Solution de l'exercice 1. — Voici le code de la figure.

```
\begin{tikzpicture}
\newcommand{\cercleA}{(0,0.67) circle (1cm)}
\newcommand{\cercleB}{(0,-0.67) circle (1cm)}
\fill[fill=blue!20] \cercleB;
\draw[color=blue,fill=white] \cercleA;
\draw[color=blue] \cercleB;
\end{tikzpicture}
```

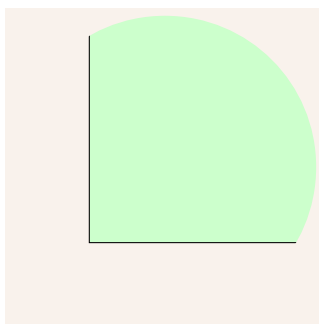
Avant de passer à la réunion et l'intersection, rappelons les commandes suivantes, vues à la séance n° 5 (voir le PDF, page 8). Tout d'abord, l'environnement `{scope}`, qui permet de délimiter l'effet de certaines commandes (notamment `\clip`) :

```
\begin{scope}
...
\end{scope}
```

La commande `\clip` s'utilise comme `\draw`, mais tout ce qui la suit ne dépassera jamais le contenu du `\clip`. Par exemple, pour dessiner dans un cercle, on fera

```
\clip (0,0) circle (2cm);
```

Un `\clip` dans un `{scope}` n'aura pas d'effet en-dehors. Voici un exemple :



```
\begin{tikzpicture}
\clip (0,0) circle (2cm);
\draw[fill=green!20] (-1,-1) rectangle (3,3);
\end{tikzpicture}
```

Exercice 2. — Utiliser les commandes `\clip` et `{scope}` pour reproduire le dessin de l'union de deux ensembles montré ci-dessus. On définira les cercles par des commandes

```
\newcommand{\cercleA}{(0,0.67) circle (1cm)}
\newcommand{\cercleB}{(0,-0.67) circle (1cm)}
```

qu'on utilisera ensuite pour le dessin. On pourra tracer d'abord la moitié haute puis la moitié basse de la figure (en *clippant* à chaque fois dans un rectangle approprié).

Solution de l'exercice 2. — Voici le code de la figure.

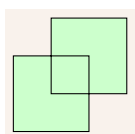
```
\begin{tikzpicture}
\newcommand{\cercleA}{(0,0.67) circle (1cm)}
\newcommand{\cercleB}{(0,-0.67) circle (1cm)}
\begin{scope}
\clip (-1.05,0) rectangle (1.05,1.7);
\draw[color=blue,fill=blue!20] \cercleA;
\end{scope}
\begin{scope}
\clip (-1.05,-1.7) rectangle (1.05,0);
\draw[color=blue,fill=blue!20] \cercleB;
\end{scope}
\end{tikzpicture}
```

Exercice 3. — Utiliser les commandes `\clip` et `{scope}` pour reproduire le dessin de l'intersection de deux ensembles montré ci-dessus. On pourra *clipper* chaque cercle à l'intérieur de l'autre.

Solution de l'exercice 3. — Voici le code de la figure.

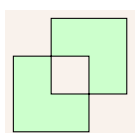
```
\begin{tikzpicture}
\newcommand{\cercleA}{(0,0.67) circle (1cm)}
\newcommand{\cercleB}{(0,-0.67) circle (1cm)}
\draw[color=blue] \cercleA;
\draw[color=blue] \cercleB;
\begin{scope}
\clip \cercleA;
\draw[color=blue,fill=blue!20] \cercleB;
\end{scope}
\begin{scope}
\clip \cercleB;
\draw[color=blue] \cercleA;
\end{scope}
\end{tikzpicture}
```

Terminons par l'exclusion. Pour cela, on va utiliser un mode de remplissage différent. Au lieu de dessiner deux objets dans deux commandes `\draw` ou `\fill` séparée, on peut choisir de les dessiner dans la même commande. Par exemple,



```
\begin{tikzpicture}
\draw[fill=green!20] (0,0) rectangle (1,1)
(0.5,0.5) rectangle (1.5,1.5);
\end{tikzpicture}
```

Noter que les bords des deux rectangles sont affichés (on peut ne pas les avoir en utilisant `\fill` à la place de `\draw`) et que l'intérieur est rempli. Si on veut que les parties communes ne soient pas remplies, on peut utiliser l'option `even odd rule` :



```
\begin{tikzpicture}
\draw[fill=green!20,even odd rule] (0,0) rectangle (1,1)
(0.5,0.5) rectangle (1.5,1.5);
\end{tikzpicture}
```

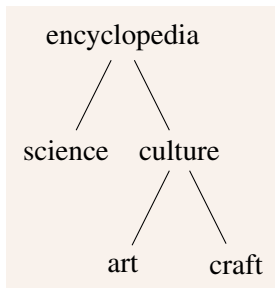
Exercice 4. — En utilisant l'option `even odd rule`, reproduire le dessin de l'exclusion de deux ensembles montré ci-dessus.

Solution de l'exercice 4. — Voici le code de la figure.

```
\begin{tikzpicture}
\newcommand{\cercleA}{(0,0.67) circle (1cm)}
\newcommand{\cercleB}{(0,-0.67) circle (1cm)}
\fill[fill=blue!20,even odd rule] \cercleA \cercleB;
\draw[color=blue] \cercleA;
\draw[color=blue] \cercleB;
\end{tikzpicture}
```

9.2 Arbres

Le but de ce § 9.2 est de dessiner des arbres, utiles par exemple en probabilités. Voici un exemple d'arbre, tiré de wikipédia.



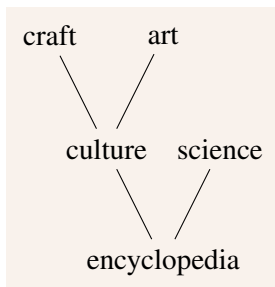
```

\begin{tikzpicture}
\node {encyclopedia}
  child {node {science}}
  child {node {culture}
    child {node {art}}
    child {node {craft}}
  }
;
\end{tikzpicture}

```

Noter que les arbres sont représentés en TikZ par un node qui peut prendre un ou plusieurs child, eux-mêmes composés de node pouvant eux-mêmes avoir des child.

Si on veut que l'arbre pousse dans une autre direction que le bas, il suffit de l'indiquer avec l'option grow. Cette option s'utilise soit avec une direction du type south east ou north, comme dans

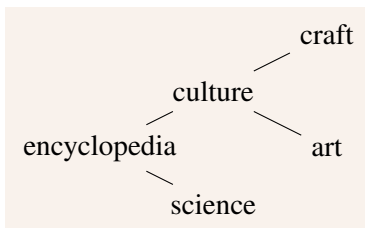


```

\begin{tikzpicture}
\node {encyclopedia} [grow=north]
  child {node {science}}
  child {node {culture}
    child {node {art}}
    child {node {craft}}
  }
;
\end{tikzpicture}

```

soit avec right ou left, comme dans

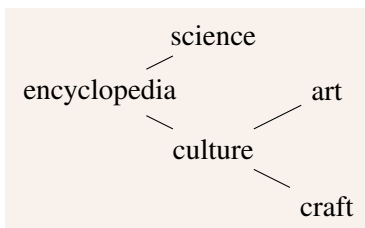


```

\begin{tikzpicture}
\node {encyclopedia} [grow=right]
  child {node {science}}
  child {node {culture}
    child {node {art}}
    child {node {craft}}
  }
;
\end{tikzpicture}

```

On peut aussi renverser l'arbre en utilisant grow' :

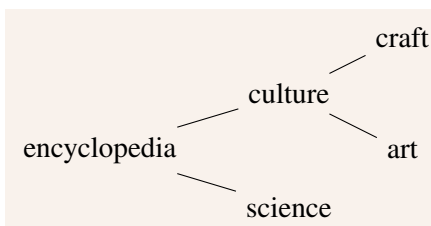


```

\begin{tikzpicture}
\node {encyclopedia} [grow'=right]
  child {node {science}}
  child {node {culture}
    child {node {art}}
    child {node {craft}}
  }
;
\end{tikzpicture}

```

Noter que dans ces deux derniers exemples, les branches sont trop rapprochées. Pour palier à cela, on peut jouer sur level distance ; par exemple,

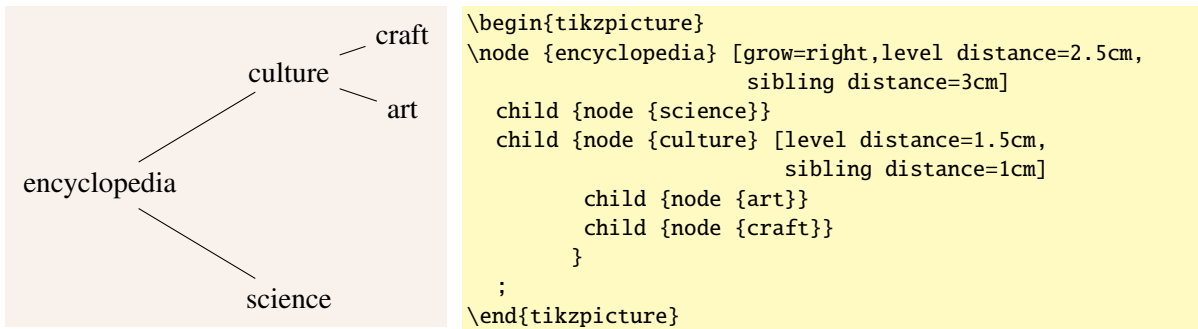


```

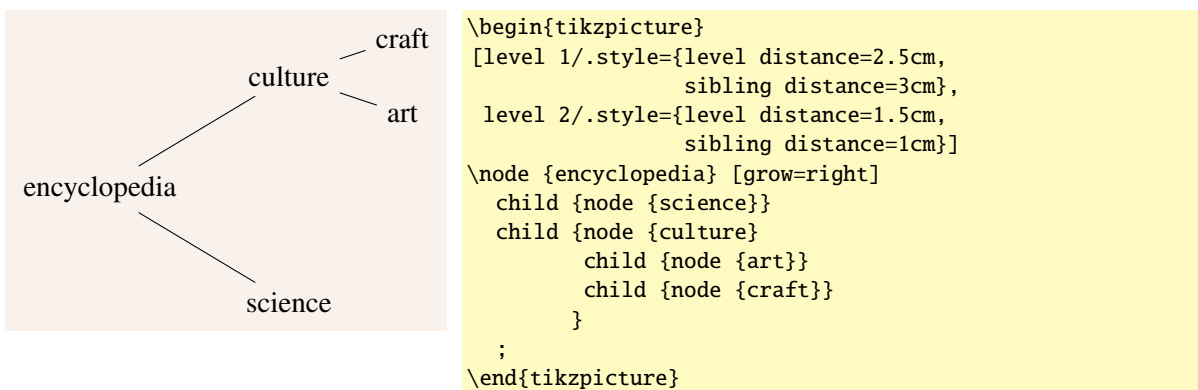
\begin{tikzpicture}
\node {encyclopedia} [grow=right,level distance=2.5cm]
  child {node {science}}
  child {node {culture} [level distance=1.5cm]
    child {node {art}}
    child {node {craft}}
  }
;
\end{tikzpicture}

```

Il peut aussi être utile de jouer sur la sibling distance (notamment pour éviter le chevauchement entre deux branches) :

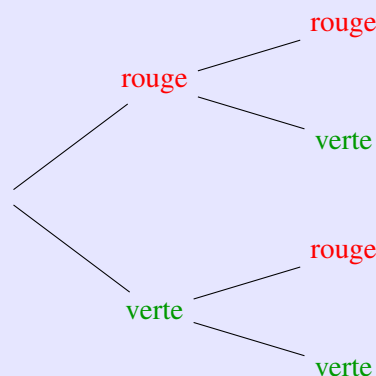


Il est également possible de fixer les paramètres de `level distance` et `sibling distance` globalement, pour toutes les branches de l'arbre en faisant

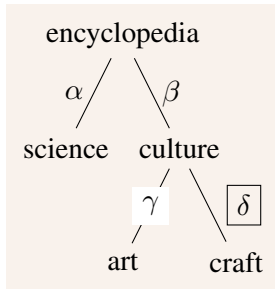


C'est surtout utile si l'arbre est fortement symétrique et a beaucoup de branches.

Exercice 5. — En utilisant les commandes vus jusqu'ici, reproduire l'arbre de probabilités suivant, associé au jeu suivant : on tire successivement deux boules dans un sac contenant 3 boules rouges et 5 boules vertes et on veut représenter l'arbre de probabilité correspondant pour visualiser les probabilités d'obtenir 2 boules rouges, 2 boules vertes et une boule de chaque couleur.



On va maintenant voir comment placer du texte sur les branches reliant deux nœuds entre elles. Il faut pour cela utiliser la clef `edge from parent`, qui s'utilise toujours *après* les `child` comme suit :

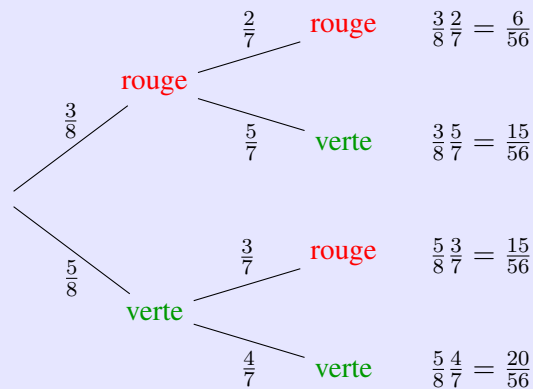


```

\begin{tikzpicture}
\node {encyclopedia}
  child {node {science}
    edge from parent node[left] {$\alpha$}
  }
  child {node {culture}
    child {node {art}
      edge from parent node[fill=white] {$\gamma$}
    }
    child {node {craft}
      edge from parent node[draw,rectangle,
        right=0.25cm] {$\delta$}
    }
    edge from parent node[right] {$\beta$}
  }
;
\end{tikzpicture}

```

Exercice 6. — Reprendre l'arbre de l'exercice précédent et y ajouter les étiquettes suivantes sur les branches.



Solution de l'exercice 6. — Voici le code de l'arbre.

```

\begin{tikzpicture}
[level 1/.style={level distance=2cm,
  sibling distance=3cm},
level 2/.style={level distance=2.5cm,
  sibling distance=1.5cm}]
\node {} [grow'=right]
  child {node {\textcolor{red}{rouge}}
    child {node {\textcolor{red}{rouge}}
      node[right=1cm] {$\frac{3}{8}\cdot\frac{2}{7} = \frac{6}{56}$}
      edge from parent node[above] {$\frac{2}{7}$}
    }
    child {node {\textcolor{darkgreen}{verte}}
      node[right=1cm] {$\frac{3}{8}\cdot\frac{5}{7} = \frac{15}{56}$}
      edge from parent node[below] {$\frac{5}{7}$}
    }
    edge from parent node[above] {$\frac{3}{8}$}
  }
  child {node {\textcolor{darkgreen}{verte}}
    child {node {\textcolor{red}{rouge}}
      node[right=1cm] {$\frac{5}{8}\cdot\frac{3}{7} = \frac{15}{56}$}
      edge from parent node[above] {$\frac{3}{7}$}
    }
    child {node {\textcolor{darkgreen}{verte}}
      node[right=1cm] {$\frac{5}{8}\cdot\frac{4}{7} = \frac{20}{56}$}
      edge from parent node[below] {$\frac{4}{7}$}
    }
    edge from parent node[above] {$\frac{5}{8}$}
  }
;
\end{tikzpicture}

```

```

    child {node {\textcolor{darkgreen}{verte}}
      node[right=1cm] {$\frac{5}{8}\cdot\frac{4}{7} = \frac{20}{56}$}
      edge from parent node[below] {$\frac{4}{7}$}
    }
    edge from parent node[below] {$\frac{5}{8}$}
  }
;
\end{tikzpicture}

```

9.3 Titre de chapitres évolués

Le but de cette section 9.3 est de montrer comment on peut utiliser TikZ en conjugaison avec `titlesec` afin de faire des titres de chapitres plus évolués. Sur ce sujet, on pourra consulter les deux ressources suivantes :

- <http://texblog.net/latex-archive/layout/fancy-chapter-tikz/>
- <http://latex-my.blogspot.com/2009/10/true-or-false-latex-can-produce-only.html>

Le but aujourd’hui est de reproduire la page suivante. L’image utilisée est

http://upload.wikimedia.org/wikipedia/commons/4/4a/Graz_University-Library_reading-room.jpg

et la bibliographie est celle faite à la séance n° 3 (document B).



Bibliographie

- [1] Anatole DURAND, *Analyses des données systémiques – Une introduction*, Masson, 1987.
- [2] Guillaume DUPONT, « Une nouvelle démonstration du théorème fondamental des analyses de données systémiques », *Journal of Systemic Data* **4** (1995), p. 456-561.
- [3] Anatole DURAND et Guillaume DUPONT, *Nouveau traité d'analyse des données systémiques*, Vuibert, 2002.

Commençons par voir comment interfacier les packages `titlesec` et `tikz`. Tout d'abord, on définit une commande, disons, `\newchaptercmd` dont l'argument est le titre de chapitre. Par exemple,

```
\newcommand{\chapterlabel}{}
\newcommand{\newchapterfont}{\Huge\bfseries\fontfamily{ugq}\selectfont}
\newcommand{\newchaptercmd}[1]{%
  \begin{center}\begin{tikzpicture}
    \node[above,draw,color=blue,fill=blue,inner sep=10pt] at (0,0)
      {\color{white}\newchapterfont\chapterlabel};
    \node[below=6pt] at (0,0) {\color{blue}\newchapterfont#1};
  \end{tikzpicture}\end{center}
}
```

Bien penser à utiliser `\chapterlabel` pour la position du numéro de chapitre. Ensuite, on utilise les commandes `\titleformat` et `\titlespacing` de la façon suivante :

```
\titleformat{\chapter}
  {\renewcommand{\chapterlabel}{\vphantom{1}\textbullet}}
  {\gdef\chapterlabel{\thechapter}}
  {0pt}
  {\newchaptercmd}
  {}
\titlespacing*{\chapter}{20pt}{0pt}{30pt}
```

(Noter l'utilisation de `\gdef` pour redéfinir *globalement* la commande `\chapterlabel` ; ne pas utiliser `\gdef` lorsque `\newcommand` et `\renewcommand` sont suffisants.) À ce stade, les titres de chapitres sont donc composés avec TikZ. On va maintenant voir les commandes TikZ nécessaires à la composition du titre de chapitre précédent.

Voici un code complet reprenant les deux jeux de commandes précédents.

```
\documentclass{report}%          autres choix : report, book

\usepackage[utf8]{inputenc}%     encodage du fichier source
\usepackage[T1]{fontenc}%       gestion des accents (pour les pdf)
\usepackage[francais]{babel}%   rajouter éventuellement english, greek, etc.
\usepackage{textcomp}%         caractères additionnels
\usepackage{amsmath,amssymb}%   pour les maths
\usepackage{lmodern}%          remplacer éventuellement par txfonts, fourier, etc.
\usepackage[a4paper]{geometry}%  taille correcte du papier
\usepackage{graphicx}%         pour inclure des images
\usepackage{xcolor}%           pour gérer les couleurs
\usepackage{microtype}%        améliorations typographiques

\usepackage{titlesec}
\usepackage{tikz}

\usepackage{hyperref}%         gestion des hyperliens
\hypersetup{pdfstartview=XYZ}%  zoom par défaut

\newcommand{\chapterlabel}{}
\newcommand{\newchapterfont}{\Huge\bfseries\fontfamily{ugq}\selectfont}
\newcommand{\newchaptercmd}[1]{%
  \begin{center}\begin{tikzpicture}
    \node[above,draw,color=blue,fill=blue,inner sep=10pt] at (0,0)
      {\color{white}\newchapterfont\chapterlabel};
    \node[below=6pt] at (0,0) {\color{blue}\newchapterfont#1};
  \end{tikzpicture}\end{center}
}
```

```

\titleformat{\chapter}
  {\renewcommand{\chapterlabel}{\vphantom{1}\textbullet}}
  {\gdef\chapterlabel{\thechapter}}
  {0pt}
  {\newchaptercmd}
  {}
\titlespacing*{\chapter}{20pt}{0pt}{30pt}

\begin{document}

\chapter{Titre}

\chapter*{Titre}

\chapter{Titre}

\end{document}

```

Voyons comment placer du matériel en référence absolue à la page. Il faut rajouter les options `[overlay,remember picture]` à `{tikzpicture}` et ensuite, on peut utiliser `current page.south` pour obtenir le sud de la page. Par exemple,

```

\begin{tikzpicture}[overlay,remember picture]
\node[above] at (current page.south) {\color{red}texte en bas de la page};
\end{tikzpicture}

```

a permis de dessiner le cadre en bas de la page. Attention, deux compilations successives sont nécessaires pour afficher le texte correctement. Si on veut placer une image en bas et non du texte, il suffirait de mettre un `\includegraphics` dans le contenu du node.

Si on veut par exemple dessiner un rectangle de 1cm de haut en haut de la page, il faut pouvoir dire à TikZ de faire un rectangle allant du nord-ouest de la page jusqu'au nord-est remonté de 1cm. Pour cela, on utilise la librairie `calc`, que l'on charge dans le préambule, juste après `\usepackage{tikz}` par

```
\usetikzlibrary{calc}
```

Ensuite, la syntaxe est la suivante :


```

\begin{tikzpicture}[overlay,remember picture]
\fill[color=blue!20] (current page.north west) rectangle
  ($(current page.north east) + (0,-1cm)$);
\end{tikzpicture}

```

L'endroit où l'on fait les calculs est toujours entre `($` et `$)` et à l'intérieur, on utilise soit des coordonnées explicites comme `(1cm,5cm)`, soit des coordonnées nommées comme `current page.north west`.

Pour faire un ovale, on peut utiliser l'option `rounded corners` de la forme `rectangle` ; par exemple,




```

\begin{tikzpicture}
\fill[color=blue!20,rounded corners] (0,0) rectangle (2,1);
\end{tikzpicture}

```

On peut spécifier au besoin l'étendu des arrondis :

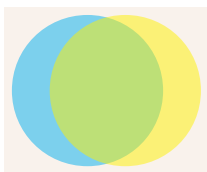


```

\begin{tikzpicture}
\fill[color=blue!20,rounded corners=0.5cm] (0,0) rectangle (2,1);
\end{tikzpicture}

```

Pour gérer l'opacité, on utilise l'option `opacity` ; par exemple,



```
\begin{tikzpicture}
\fill[color=cyan,opacity=0.5] (0,0) circle (1cm);
\fill[color=yellow,opacity=0.5] (0.5,0) circle (1cm);
\end{tikzpicture}
```

La force de l'opacité est un nombre compris entre 0 et 1.

Exercice 7. — En utilisant les commandes précédentes, reproduire la page 8.

Solution de l'exercice 7. — Voici le code

```
\documentclass{report}

\usepackage[français]{babel}

\usepackage{graphicx}
\usepackage[a4paper]{geometry}

\usepackage{calc}

\usepackage{titlesec}

\usepackage{lipsum}

\usepackage{tikz}
\usetikzlibrary{calc}

\newcommand{\nomproprebib}[2]{#1 \textsc{#2}}
\newcommand{\entreebiblivre}[4]{#1, \emph{#2}, #3, #4.}
\newcommand{\entreebibarticle}[7]{#1, \og #2\fg{, \emph{#3} \textbf{#4} (#5), p.~#6-#7.}

\newcommand{\ifempty}[3]{\ifx#1\empty#2\else#3\fi}

\newcommand{\thechapterimage}{}
\newcommand{\chapterimage}[1]{\renewcommand*{\thechapterimage}{#1}}

\newcommand{\chapterheadfont}{\fontfamily{ugq}\fontsize{24pt}{30pt}\selectfont}
\newcommand{\newchaptercmd}[1]{%
\begin{tikzpicture}[overlay,remember picture]
\begin{scope}
\clip (current page.north west) rectangle ($ (current page.north east) + (0cm,-12cm)$);
\fill[blue!20] (current page.north west) rectangle
($ (current page.north east) + (0cm,-12cm)$);
\node[below=-0.35cm] at (current page.north)
{\ifempty{\thechapterimage}{}{\includegraphics[width=\paperwidth]{\thechapterimage}}};
\end{scope}
\node[right=2cm,rectangle,draw,color=white,fill=white,opacity=0.7,inner ysep=10pt,
inner xsep=15pt,rounded corners=0.75cm] at ($ (current page.north west) + (0cm,-7cm)$)
{\chapterheadfont\phantom{pb}\phantom{#1}\hspace{\paperwidth}\null};
\node[right=2cm,inner ysep=10pt,inner xsep=15pt] at ($ (current page.north west) + (0cm,-7cm)$)
{\chapterheadfont\phantom{pb}\color{red!60!black}\chapterlabel#1};
\end{tikzpicture}
}

\titleformat{\chapter}
```

```
{\gdef\chapterlabel{}}
{\gdef\chapterlabel{\thechapter.\ }}
{0pt}
{\newchaptercmd}
{}
\titlespacing*{\chapter}{0pt}{0pt}{8cm}

\begin{document}

\chapterimage{}
\chapter{Introduction}

\lipsum

\chapterimage{Graz_University-Library_reading-room}
\begin{thebibliography}{1}
  \bibitem{Durand.systemique} \entreebiblivre{\nomproprebib{Anatole}{Durand}}
    {Analyses des données systémiques -- Une introduction}{Masson}{1987}
  \bibitem{Dupont.nouvelle.demo} \entreebibarticle{\nomproprebib{Guillaume}{Dupont}}
    {Une nouvelle démonstration du théorème fondamental des analyses de données
    systémiques}{Journal of Systemic Data}{4}{1995}{456}{561}
  \bibitem{Durand.Dupont.systemique} \entreebiblivre{\nomproprebib{Anatole}{Durand}
    et \nomproprebib{Guillaume}{Dupont}}{Nouveau traité d'analyse des données
    systémiques}{Vuibert}{2002}
\end{thebibliography}

\end{document}
```