

Sesammanuel Documentation

Documentation pour l'utilisation de la classe \LaTeX `sesamanuel.cls` et des packages `sesamanuel.sty` et `sesamanuelTIKZ.sty` de *Sésamath*.

Partie 1 : Introduction page 3

Cette partie contient une présentation de la classe : historique, méthode d'installation, méthode de compilation.

Partie 2 : Classe pour un ouvrage page 7

Si vous souhaitez utiliser la structure complète de la classe pour composer un manuel (un ou plusieurs chapitres accompagnés d'annexes), vous trouverez dans cette partie, les explications de toutes les commandes créées par cette classe, leur fonctionnement et les possibilités de personnalisation.

Partie 3 : Classe pour quelques feuilles page 47

Si vous souhaitez intégrer quelques extraits choisis du manuel *Sésamath 2^{de}* dans des documents sans utiliser la classe complète, vous trouverez votre bonheur dans cette partie.

Partie 4 : Détails Typographiques page 51

Vous trouverez dans cette partie, les commandes spéciales utilisées dans le manuel *Sésamath 2^{de}* pour homogénéiser certaines présentations.

Partie 5 : Figures TIKZ page 65

Vous trouverez dans cette partie, les commandes spéciales utilisées dans le manuel *Sésamath 2^{de}* pour homogénéiser les figures.

Introduction

Constructions offertes par la classe

Dans ce premier chapitre, nous allons présenter quelques généralités sur la classe `sesammanuel` :

- ▶ sa raison d'être,
- ▶ son installation et
- ▶ quelques détails techniques.

Les problèmes d'installations sont maintenant souvent résolus grâce à des procédures d'installation automatisées (T_EXLive ou MikT_EX)(section 2).

En revanche la section 3 est réellement importante à lire puisque la compilation actuellement la plus fréquemment utilisée (à savoir celle par `pdflatex`) ne fonctionnera pas avec cette classe.



Auto-évaluation

Des ressources numériques pour préparer le chapitre sur manuel.sesamath.net



1 Pourquoi ne faut-il pas passer ce chapitre sans en lire le contenu ?

2 Comment utiliser cette classe pour composer un seul exercice ?

3 Je n'aime pas la façon avec laquelle la classe com-

pose un ouvrage. Comment faire ?

4 Pour adapter la classe à mes demandes, puis-je modifier le code source ?

▶▶▶ Voir solutions p. 74



1. Présentation

■ Historique

La classe `sesamaneul` a été une commande de l'association française *Sésamath*. Cette classe a servi à éditer le manuel *Sésamath* 2^{de}.

L'association *Sésamath* a souhaité que la classe utilisée pour produire ce document soit mise à la disposition de tous *via* son site <http://www.sesamath.net/> et en le déposant sur CTAN sous licence libre.

■ Principe

Pour que la classe soit utilisable facilement par tout un chacun, même s'il s'agit d'une classe conçue pour fabriquer un ouvrage par des professionnels, il a été décidé, dès le départ, que toute la chaîne de production TEX n'utiliserait que des produits courants.

En particulier, la maquette ne fait appel à aucune fonte non présente par défaut dans toute distribution TEX ni a aucune extension qui ne se trouve pas sur CTAN et sur les distributions classiques. Le tableau 1.1 donne la liste des extensions et fontes utilisées par la classe.

En réalité, **la classe a un double but** :

- composer un ouvrage en respectant une maquette bien précise d'un éditeur ;
- permettre à n'importe qui de mettre en page tout ou partie de cet ouvrage.

L'idée est que des enseignants de mathématiques puissent reprendre une partie du code source des ouvrages, de pouvoir les modifier éventuellement et de produire des feuilles séparées ou bien des documents à projeter en classe.

Donc, la classe assure de pouvoir composer un manuel en respectant un cahier des charges précis et de pouvoir également se soustraire à certaines exigences de ce cahier des charges en gardant une syntaxe commune pour les fichiers sources.

Plus précisément, l'idée est de construire un ouvrage papier dont le source est éclaté en autant de fichiers sources qu'il y a d'éléments dans cet ouvrage (cours, exercice, TP, corrigés, etc.) et de pouvoir reconstruire un document en gardant les éléments que l'on souhaite (un seul exercice, un choix d'exercices, une partie cours, des activités, un thème complet...).

Ces deux exigences font que la classe peut être comprise comme ayant deux modes de fonctionnement différents et c'est comme cela que les fonctionnalités seront présentées.





TABLE 1.1 – Liste des extensions utilisées par la classe

Extension	Version et date
classe book	v1.4h 2007/10/19
etex	v2.0 1998/03/26
fontenc	v1.99g 2005/09/27
inputenc	v1.1d 2008/03/30
fontspec	
xunicode	
helvet	v9.2a 2005/04/12
mathpazo	v9.2a 2005/04/12
mathrsfs	v1.0 1996/01/01
pifont	v9.2a 2005/04/12
eurosym	v1.1 1998/08/06
geometry	v5.6 2010/09/12
ifpdf	v2.3 2011/01/30
crop	v1.9 2003/05/20
ifmtarg	v1.2a 2009/09/02
mathtools	v1.13 2013/02/12
amssymb	v3.01 2013/01/14
longtable	v4.11 2004/02/01
tabularx	v2.07 1999/01/07
multitrow	V1.6 2004/05/05
xcolor	v2.11 2007/01/21
pst-all	voir note
pstricks-add	voir note
fancyhdr	v3.2 2005/03/22
fancyvrb	v2.7a 2008/02/07
multicol	v1.7a 2011/06/27
babel	v3.9h 2013/12/03
numprint	v1.39 2012/08/20

NOTE : la bibliothèque d’extensions PSTricks contient une vingtaine d’extensions dont les plus récentes datent de février 2014. La classe `sesamanuel` n’utilise cependant aucune fonctionnalité de PSTricks définie après 2009.



2. Installation

■ Si vous avez une version récente : **T_EXLive 2013 (à jour), de MikT_EX ou MacT_EX**

La classe se trouvant sur CTAN, la classe est déjà installée et vous n'avez rien à faire.

■ Si vous avez une version ancienne : **distribution T_EX avant 2013**

Il faudra installer la classe « à la main ». C'est très simple, pour que cela fonctionne, il n'y a besoin que du seul fichier `sesamanuel.cls`. Il faut également que la compilation retrouve ce fichier sur votre disque dur.

- Il y a une première solution peu élégante qui consiste à placer ce fichier dans le même répertoire que le document maître.

C'est peu élégant car il faudra dupliquer ce fichier dans autant d'endroits qu'il y aura de répertoires de travail... et lorsque la version de la classe changera (par exemple pour corriger un bug), il faudra alors répercuter ce changement dans tous ces répertoires.

- Une deuxième solution plus élégante consiste à placer le fichier dans l'arborescence principale de T_EX. La racine de celle-ci dépend du système d'exploitation et de la distribution T_EX utilisée.

- Sous MikT_EX ou T_EXLive sous Windows, la racine peut être `C:\texmf\`.

- Sous T_EXLive sous Linux, la racine peut être `/usr/share/texlive/texmf-dist/` pour une distribution installée par le système ou bien `/usr/local/texlive/<année>/texmf-dist` pour une distribution installée *via* l'installeur T_EXLive.

- Sous MacT_EX, la racine se trouve normalement en `/usr/local/texlive/<année>/texmf-dist`.

3. Mode de compilation

Les documents produits avec la classe ne demandent pas d'outils externes pour être construits. En particulier, le lexique ne demande pas de passage par `makeindex` ou bien `texindy` ou `xindy` pour que ses entrées soient classées.

REMARQUE : La classe utilise intensément PSTricks pour tout ce qui est graphique. Cela implique qu'une compilation avec `pdflatex` ne fonctionnera pas. Il y a deux solutions :

- une compilation avec la chaîne `latex → dvips → ps2pdf` ou bien des programmes équivalents transformant le dvi en ps (ou pdf) ;
- une compilation avec `xelatex`

La classe détecte automatiquement le mode de compilation et adapte ses appels d'extension au moteur utilisé.

4. Option de compilation

Lorsqu'on compose un manuel avec la maquette `sesamath`, on dispose des options de classes suivants :

- Tous les codages d'entrées que connaît le package `inputenc`. L'option par défaut est `utf8`.
- `crop` ou `nocrop` pour obtenir, ou non, les repères photographiques de coupe. L'option par défaut est `crop`.



Classe pour un ouvrage

Les différentes parties d'un ouvrage

Les différentes parties d'un ouvrage vont être présentées en détail dans les pages qui suivent. Voici la liste de tous les types de parties qui vont composer chaque chapitre ainsi que les différents types d'annexes que l'on rencontre.

- ▶ Organisation générale
- ▶ Première page d'un chapitre
- ▶ Activités d'approche
- ▶ Cours-Méthodes
- ▶ Exercices d'entraînement
- ▶ Exercices d'approfondissement
- ▶ Récréation, énigmes

- ▶ Je teste mes connaissances
- ▶ Travaux pratiques
- ▶ Annexes générales
- ▶ Liste des méthodes
- ▶ Solutions
- ▶ Propriétés
- ▶ Lexique

Avant de voir toutes ces parties, on listera toutes les couleurs, les longueurs, les commandes de changement de fontes et les textes fixes utilisés par la maquette et définis par la classe.



Auto-évaluation

Des ressources numériques pour préparer le chapitre sur manuel.sesamath.net



1 Serez-vous capable de lire ce manuel en entier en moins de deux heures ?

2 Le source de ce manuel est normalement à votre disposition (fichier `sesamanuel-doc-fr.tex`).

Repérez cette première page et changez l'encadré « Auto-évaluation » en « Exercices pour auteurs ».

3 Pourquoi l'exercice précédent est-il un peu idiot (trop facile) si on le fait vraiment avec le source du manuel ? Essayez de répondre à cet exercice sans regarder ni la solution, ni le source du manuel !

▶▶▶ Voir solutions p. 74



1. Organisation générale

■ Principe

L'ouvrage suit une maquette. La suite des chapitres est répartie en trois thèmes suivis d'annexes. Trois thèmes sont proposés : dans l'ordre, Statistiques et probabilités, Fonctions, Géométrie. Ces thèmes sont introduits par les macros respectives :

1) `\themaSP` 2) `\themaF` 3) `\themaG`

Le fait d'être dans un thème va faire que certains éléments vont avoir une couleur particulière. Les titres de chapitres seront accompagnés d'une lettre correspondant au thème et la liste des méthodes sera classée par thème.

MÉTHODE 1 Créer de nouveaux thèmes

La classe offre la possibilité de créer de nouveaux thèmes et de changer toutes les couleurs liées au thème grâce à la commande `\NewThema`. Cette commande demande sept arguments obligatoires qui vont être, dans l'ordre :

- lettre après `\thema` (normalement en capitale) ;
- lettre en minuscule ;
- titre du thème en minuscule ;
- titre du thème avec initiale en capitale ;
- titre du thème en capitale ;
- couleur du cadre pour le rectangle vertical de chapitre ainsi que celle du dernier carré en bas à gauche ;
- couleur de l'avant-dernier carré en bas à gauche du rectangle vertical de chapitre ainsi que celle du rectangle de pied de page ;

Exercice d'application Pour cette documentation, on a créé un thème « manuel »

Correction La commande est la suivante :

```
\NewThema{M}{m}{manuel}{Manuel}{MANUEL}{PartieFonction}{A3}
```

Les deux syntaxes de couleurs `PartieFonction` et `A3` vont être expliquée immédiatement après.

A. Couleurs

La classe offre la possibilité de modifier les couleurs.

Il faut être conscient que changer les couleurs (ou les autres éléments) fera qu'on ne respectera plus la maquette.

La maquette définit un nuancier avec lequel seront définies toutes les autres couleurs. La table 2.1 page suivante donne la liste de toutes les couleurs de ce nuancier avec leur nom et leur définition (CMYK sauf pour les teintes de gris). Certaines couleurs de ce nuancier ne servent pas dans le résultat final. La maquette a évolué mais comme cela ne posait aucun problème, le nuancier initial a été gardé.

À partir de ce nuancier, la classe va définir plus de 200 couleurs employées dans l'ensemble d'un ouvrage. Le (grand) tableau 2.2 pages 10-16 donne la liste de ces couleurs avec leur nom, leur définition initiale et leur signification. Pour mieux s'y retrouver, on a organisé ce tableau en parties correspondant à celles de l'ouvrage.



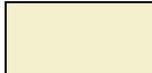
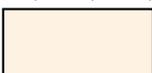
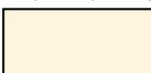
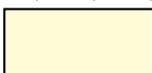
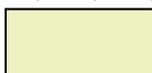
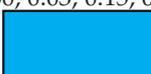
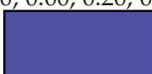
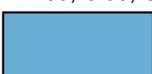
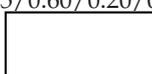
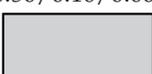
			
A1 : 1.00, 0.00, 0.00, 0.50	A2 : 0.60, 0.00, 0.00, 0.10	A3 : 0.30, 0.00, 0.00, 0.05	A4 : 0.10, 0.00, 0.00, 0.00
			
B1 : 0.00, 1.00, 0.60, 0.40	B2 : 0.00, 0.85, 0.60, 0.15	B3 : 0.00, 0.20, 0.15, 0.05	B4 : 0.00, 0.05, 0.05, 0.00
			
C1 : 0.00, 1.00, 0.00, 0.50	C2 : 0.00, 0.60, 0.00, 0.20	C3 : 0.00, 0.30, 0.00, 0.05	C4 : 0.00, 0.10, 0.00, 0.05
			
D1 : 0.00, 0.00, 1.00, 0.50	D2 : 0.20, 0.20, 0.80, 0.00	D3 : 0.00, 0.00, 0.20, 0.10	D4 : 0.00, 0.00, 0.20, 0.05
			
F1 : 0.00, 0.80, 0.50, 0.00	F2 : 0.00, 0.40, 0.30, 0.00	F3 : 0.00, 0.15, 0.10, 0.00	F4 : 0.00, 0.07, 0.05, 0.00
			
G1 : 1.00, 0.00, 0.50, 0.00	G2 : 0.50, 0.00, 0.20, 0.00	G3 : 0.20, 0.00, 0.10, 0.00	G4 : 0.10, 0.00, 0.05, 0.00
			
H1 : 0.40, 0.00, 1.00, 0.10	H2 : 0.20, 0.00, 0.50, 0.05	H3 : 0.10, 0.00, 0.20, 0.00	H4 : 0.07, 0.00, 0.15, 0.00
			
J1 : 0.00, 0.50, 1.00, 0.00	J2 : 0.00, 0.20, 0.50, 0.00	J3 : 0.00, 0.10, 0.20, 0.00	J4 : 0.00, 0.07, 0.15, 0.00
			
FondOuv : 0.00, 0.05, 0.10, 0.00	FondAutoEvaluation : 0.00, 0.03, 0.15, 0.00	FondTableaux : 0.00, 0.00, 0.20, 0.00	FondAlgo : 0.07, 0.00, 0.30, 0.00
			
BleuOuv : 1.00/0.00/0.00/0.00	PartieFonction : 1.00/0.00/0.00/0.00	PartieGeometrie : 0.80/0.80/0.00/0.00	PartieStatistique : 0.95/0.60/0.20/0.00
			
U1 : 0.50/0.10/0.00/0.10	U2 : 0.20/0.15/0.00/0.00		Blanc : gray 100%
			
Gris1 : gray 80%	Gris2 : gray 60%	Gris3 : gray 40%	Noir : gray 0%

TABLE 2.1 – Nuancier de la classe



Couleurs dans tout le document

AlgoLineColor	F3	Couleur des lignes du logo \algo : ALGO
AlgoBkgColor	F1	Couleur de fond du logo \algo
AlgoTextColor	Blanc	Couleur du texte du logo \algo
TiceLineColor	A3	Couleur des lignes du logo \tice : INFO
TiceBkgColor	A1	Couleur de fond du logo \tice
TiceTextColor	Blanc	Couleur du texte du logo \tice
CadreLineColor	D3	Couleur par défaut des cadres avec coin biseauté
CadreBkgColor	Blanc	Couleur par défaut du fond des cadres avec coin biseauté
AlgorithmeRuleColor	D2	Couleur du bord du cadre d'un algorithme
AlgorithmeBkgColor	FondAlgo	Couleur du fond du cadre d'un algorithme
CouleurAlgoAffiche	Noir	Couleur du texte Afficher la valeur et Afficher de \AfficherVar et de \Afficher
CouleurAlgoDemander	Noir	Couleur du texte Demander de \Demander
CouleurAlgoSaisir	Noir	Couleur du texte Saisir de \Saisir
CouleurAlgoStocker	Noir	Couleur du texte Stocker de \Stocker
CouleurAlgoSi	Noir	Couleur du texte Si , Sinon , Alors et Fin Si de \sialors et \sialorssinon
CouleurAlgoTantque	Noir	Couleur du texte Tant que et Fin Tant que de \tantque
CouleurAlgoPour	Noir	Couleur du texte Pour , variant de , à , avec un pas de , faire et Fin Pour de \pour et \pourAvecPas
CouleurAlgoCalcule	Noir	Couleur du texte Calculer de \Calculer
CouleurAlgoInitialiser	Noir	Non utilisé
CouleurAlgoIncrementer	Noir	Non utilisé
CouleurAlgoCommentaire	Noir	Couleur du texte de commentaire de \DeclareVar
CouleurAlgoDonnerValeur	Noir	Couleur du texte pour le texte de \DonnerValeur
CommentaireItemColor	Noir	Couleur des items dans l'environnement commentaire
FiletTableauColor	Gris3	Couleur des filets des tableaux
FondSudokuColor	FondTableaux	Couleur de fond pour les cases colorées des sudokus

Première page d'un chapitre

ChapterTopFrameColor	*	Couleur du rectangle supérieur horizontal (sur la largeur de page) dans l'en-tête de chapitre
ChapterBottomFrameColor	J4	Couleur du rectangle inférieur horizontal (sur la largeur de page) dans l'en-tête de chapitre
ThemaTopFrameColor	J1	Couleur du rectangle supérieur vertical (où il y a le titre du thème) dans l'en-tête de chapitre
ThemaBottomFrameColor	J2	Couleur du rectangle inférieur vertical (où il y a le titre du thème) dans l'en-tête de chapitre
ThemaTitleColor	Blanc	Couleur du titre du thème en haut du rectangle vertical
ChapterNumBkgColor	H2	Couleur de fond du rectangle au-dessous du numéro de chapitre
ChapterNumSquare0Color	Blanc	Couleur de texte dans les zones qui utilisent les petits carrés en dégradé





ChapterNumSquare1Color	H1	Couleur de dégradé des petits carrés (voir note **)
ChapterNumSquare2Color	H2	Couleur de dégradé des petits carrés (voir note **)
ChapterNumSquare3Color	H4	Couleur de dégradé des petits carrés (voir note **)
ChapterNumSquare4Color	*	Couleur de dégradé des petits carrés (voir note **)
ChapterNumSquare5Color	*	Couleur de dégradé des petits carrés (voir note **)
FootTitleHeadColor	*	Couleur du titre de chapitre dans les pieds de page de tout le chapitre
FirstChapterFootColor	*	Couleur du rectangle de pied de page pour la première page du chapitre
ChapterTitleColor	B1	Couleur du titre de chapitre en première page
ChapAppColor	Blanc	Non utilisé
ChapterNumColor	Blanc	Couleur du numéro de chapitre
ChapterNumFrameColor	*	Couleur de fond du numéro de chapitre
PrerequisTitleColor	B2	Couleur du titre des prérequis
PrerequisBkgColor	A4	Couleur de fond du cadre des prérequis
PrerequisItemColor	B2	Couleur des étiquettes d'item des prérequis
AEItemColor	B2	Couleur des étiquettes d'item dans l'auto-évaluation
AETitleFrame0Color	Blanc	Couleur du titre de l'auto-évaluation (voir note **)
AETitleFrame1Color	B2	Couleur des cercles dégradés (voir note **)
AETitleFrame2Color	B3	Couleur des cercles dégradés (voir note **)
AETitleFrame3Color	A4	Couleur des cercles dégradés (voir note **)
AETitleFrame4Color	A3	Couleur des cercles dégradés (voir note **)
AETitleFrame5Color	A2	Couleur des cercles dégradés (voir note **)
AETitleColor	Blanc	Couleur du titre de l'auto-évaluation
AEExoNumColor	Blanc	Couleur du numéro des exercices dans l'auto-évaluation
AEExoNumFrameColor	A2	Couleur du cadre des numéros des exercices dans l'auto-évaluation
AEFrameColor	FondAutoEvaluation	Couleur du cadre général des exercices de l'auto-évaluation
AECartoucheCorrBkgColor	J2	Couleur de fond du cartouche de référence à la page de correction
AECartoucheCorrArrowColor	G2	Couleur des flèches dans le cartouche de référence aux corrections
AECartoucheCorrVCPCColor	B2	Couleur du texte <code>\StringVoirCorriges</code>

Activités d'approche

ActiviteHeadFrame0Color	Blanc	Couleur du titre <code>\StringActivitesApproche</code> (voir note **)
ActiviteHeadFrame1Color	C1	Couleur des carrés dégradés (voir note **)
ActiviteHeadFrame2Color	C2	Couleur des carrés dégradés (voir note **)
ActiviteHeadFrame3Color	C3	Couleur des carrés dégradés (voir note **)
ActiviteHeadFrame4Color	D3	Couleur des carrés dégradés (voir note **)
ActiviteHeadFrame5Color	G1	Couleur des carrés dégradés (voir note **)



ActiviteFootColor	D2	Couleur du rectangle de pied de page
FootChapterNumColor	U1	Non utilisé
ActiviteSubtitleColor	H1	Couleur du titre de l'activité en cours
ActiviteNumColor	Blanc	Numéro de l'activité
ActiviteDebatTopColor	G4	Couleur dans la partie supérieure du losange des débats
ActiviteDebatBottomColor	D4	Couleur dans la partie inférieure du losange des débats
ActiviteActiviteTopColor	D3	Couleur dans la partie supérieure du losange des activités
ActiviteActiviteBottomColor	G2	Couleur dans la partie inférieure du losange des activités
ActiviteTitleColor	C1	Titre de l'activité débats (soit Activité soit Débat)
ActiviteItemColor	C1	Couleur des étiquettes des item dans les activités et débats
ActivitePartieColor	A2	Couleur du texte des parties dans les activités débats

Cours-Méthodes

CoursHeadFrame0Color	Blanc	Couleur du texte de l'en-tête (voir note **)
CoursHeadFrame1Color	J1	Couleur des carrés dégradés (voir note **)
CoursHeadFrame2Color	J2	Couleur des carrés dégradés (voir note **)
CoursHeadFrame3Color	J4	Couleur des carrés dégradés (voir note **)
CoursHeadFrame4Color	A3	Couleur des carrés dégradés (voir note **)
CoursHeadFrame5Color	A2	Couleur des carrés dégradés (voir note **)
CoursFootColor	A3	Couleur du rectangle de pied de page
SectionFrame1Color	J4	Couleur de fond du carré supérieur droit du numéro de section
SectionFrame2Color	J2	Couleur de fond du rectangle gauche et du rectangle inférieur du numéro de section
SectionFrame3Color	J1	Couleur de fond du carré inférieur gauche du numéro de section
SectionNumColor	A1	Couleur du numéro de section
SectionTitleColor	A1	Couleur du titre de section
SubsectionNumColor	B2	Couleur du numéro de sous-section
SubsectionTitleColor	J1	Couleur du titre de sous-section
DefSquareColor	B2	Couleur du carré initial des environnements de type définition
DefTitleColor	A1	Couleur du titre des environnements de type définition (c'est-à-dire « DÉFINITION », « PROPRIÉTÉ », etc)
DefSubtitleColor	B2	Couleur du sous-titre des environnements de type définition (donné par l'utilisateur)
DefItemColor	B2	Couleur des étiquettes des items dans les environnements de type définition
DefFrameColor	J3	Couleur de fond du cadre du contenu des environnements de type définition





RemTitleColor	H1	Couleur des titres des environnements de type remarque (Remarque(s), Notation(s), Vocabulaire)
RemItemColor	H1	Couleur des étiquettes des items dans les environnements de type remarque
ExempleRuleColor	J1	Couleur des filets dans les environnements de type exemple (exemple avec correction)
ExempleEdgeFrameColor	J2	Couleur des bords gauche et inférieur des cartouches exemple et correction
ExempleBkgFrameColor	F4	Couleur de fond des cartouches exemple et correction
ExempleTitleColor	A1	Couleur des titre exemple et correction
ExempleItemColor	J1	Couleur des étiquettes des items dans les environnements de type exemple
MethodeTitleFrameColor	A2	Couleur du fond pour le titre de la méthode et son numéro
MethodeTitleColor	Blanc	Couleur du titre de la méthode (MÉTHODE) et de son numéro
MethodeSubtitleColor	B2	Couleur du sous-titre de la méthode (donnée par l'utilisateur)
MethodeIntroBkgFrameColor	A4	Couleur du fond du texte introductif de la méthode
ExAppEdgeFrameColor	A2	Couleur du trait bas du cartouche Exercice d'application
ExAppBkgFrameColor	H1	Couleur de fond du cartouche Exercice d'application
ExAppTitleColor	Blanc	Couleur du titre Exercice d'application
ExAppCorrEdgeFrameColor	H1	Couleur du trait bas du cartouche Correction
ExAppCorrBkgFrameColor	A2	Couleur de fond du cartouche Correction
ExAppCorrTitleColor	Blanc	Couleur du titre Correction
ExAppItemColor	A2	Couleur des étiquettes pour les items dans les exercices d'application (et les correction)
MethodeRuleColor	A3	Couleur des filets verticaux (large ou étroit) dans les méthodes
ProofRuleColor	B1	Couleur du filet pour les preuves
ProofTitleColor	A1	Couleur du titre des preuves
ProofTopFrameColor	B4	Couleur de fond du cadre de titre des preuves
ProofBottomFrameColor	B3	Couleur du trait inférieur de titre des preuves
ProofTriangleFrameColor	B1	Couleur du triangle supérieur gauche du titre des preuves
ProofItemColor	J2	Couleur des étiquettes des items dans les preuves
Exercices d'entraînement		
ExoBaseHeadFrame0Color	Blanc	Couleur du texte de l'en-tête (voir note **)
ExoBaseHeadFrame1Color	G1	Couleur des carrés dégradés (voir note **)
ExoBaseHeadFrame2Color	G2	Couleur des carrés dégradés (voir note **)
ExoBaseHeadFrame3Color	G4	Couleur des carrés dégradés (voir note **)
ExoBaseHeadFrame4Color	J2	Couleur des carrés dégradés (voir note **)



ExoBaseHeadFrame5Color	J1	Couleur des carrés dégradés (voir note **)
ExoBaseFootColor	J2	Couleur du rectangle de pied de page
ExerciceColumnRuleColor	J1	Couleur du trait de séparation dans les exercices et les corrigés d'exercice en double colonne
ExercicesTitleColor	C1	Couleur du titre d'une série d'exercices
ExercicesTitleVRuleColor	G2	Couleur pour la barre verticale du titre d'une série d'exercices
ExercicesTitleHRuleColor	C3	Couleur pour la barre horizontale du titre d'une série d'exercices
ExercicesTitleSquareColor	J1	Couleur pour le carré d'intersection des deux barres
ExerciceNumColor	Blanc	Couleur du numéro d'exercice
ExerciceTitleColor	F1	Couleur du titre d'un exercice
ExerciceNumFrameColor	G1	Couleur du cadre du numéro d'un exercice non corrigé
ExerciceCorrigeNumFrameColor	J1	Couleur du cadre du numéro d'un exercice corrigé
ExerciceItemColor	G1	Couleur des puces
ExercicePartieColor	J1	Couleur des parties dans un exercice
ExerciceRefMethodeColor	A2	Couleur du cadre de référence à une méthode dans un exercice

Exercices d'approfondissement

ExoApprHeadFrame0Color	Blanc	Couleur du texte de l'en-tête (voir note **)
ExoApprHeadFrame1Color	G1	Couleur des carrés dégradés (voir note **)
ExoApprHeadFrame2Color	G2	Couleur des carrés dégradés (voir note **)
ExoApprHeadFrame3Color	G4	Couleur des carrés dégradés (voir note **)
ExoApprHeadFrame4Color	J2	Couleur des carrés dégradés (voir note **)
ExoApprHeadFrame5Color	J1	Couleur des carrés dégradés (voir note **)
ExoApprFootColor	J2	Couleur du rectangle de pied de page

Je teste mes connaissances

ConnHeadFrame0Color	Blanc	Couleur du texte de l'en-tête (voir note **)
ConnHeadFrame1Color	A2	Couleur des carrés dégradés (voir note **)
ConnHeadFrame2Color	A3	Couleur des carrés dégradés (voir note **)
ConnHeadFrame3Color	J2	Couleur des carrés dégradés (voir note **)
ConnHeadFrame4Color	J1	Couleur des carrés dégradés (voir note **)
ConnHeadFrame5Color	F1	Couleur des carrés dégradés (voir note **)
ConnFootColor	J2	Couleur du rectangle de pied de page
AcquisTitleColor	G1	Couleur du titre « À la fin du chapitre, je dois être capable de »
AcquisBkgColor	A4	Couleur de fond du titre
AcquisItemColor	F1	Couleur des puces dans le cadre des acquis
QCMAEFrameColor	G1	Couleur du cadre des QCM dans l'auto-évaluation
QCMAETitleCircle1Color	J1	Couleur des cercles concentriques dans le bandeau de l'auto-évaluation
QCMAETitleCircle2Color	J2	Couleur des cercles concentriques dans le bandeau de l'auto-évaluation



QCMAETitleCircle3Color	F1	Couleur des cercles concentriques dans le bandeau de l'auto-évaluation
QCMAETitleColor	Blanc	Couleur du titre du bandeau de l'auto-évaluation
QCMIItemNumColor	G1	Couleur du numéro encerclé des QCM
QCMFrameColor	J1	Couleur du cadre pour la référence à un QCM
QCMTextFrameColor	Blanc	Couleur du texte pour la référence à un QCM
QCMExoNumColor	Blanc	Couleur pour le numéro de QCM
QCMExoNumFrameColor	J1	Couleur pour le cadre du numéro de QCM
QCMLineColor	A4	Couleur pour les lignes encadrant les QCM
Travaux pratiques, récréation, énigmes		
TPHeadFrame0Color	Blanc	Couleur du texte de l'en-tête (voir note **)
TPHeadFrame1Color	H1	Couleur des carrés dégradés (voir note **)
TPHeadFrame2Color	H2	Couleur des carrés dégradés (voir note **)
TPHeadFrame3Color	H3	Couleur des carrés dégradés (voir note **)
TPHeadFrame4Color	C3	Couleur des carrés dégradés (voir note **)
TPHeadFrame5Color	C2	Couleur des carrés dégradés (voir note **)
TPFootColor	C3	Couleur du rectangle de pied de page
TPTopColor	J3	Couleur de la partie supérieure du cadre du numéro de TP
TPBottomColor	C3	Couleur de la partie inférieure du cadre du numéro de TP
TPTitleColor	H1	Couleur du titre du TP
TPPartieColor	J1	Couleur des titres de parties dans les TP
TPPartieBkgColor	C2	Couleur du cadre de numéro de partie dans les TP
TPPartieNumColor	Blanc	Couleur du numéro de partie dans les TP
RecreationHeadFrame0Color	Blanc	Couleur du texte de l'en-tête (voir note **)
RecreationHeadFrame1Color	J1	Couleur des carrés dégradés (voir note **)
RecreationHeadFrame2Color	J2	Couleur des carrés dégradés (voir note **)
RecreationHeadFrame3Color	H4	Couleur des carrés dégradés (voir note **)
RecreationHeadFrame4Color	H2	Couleur des carrés dégradés (voir note **)
RecreationHeadFrame5Color	H1	Couleur des carrés dégradés (voir note **)
RecreationFootColor	H2	Couleur du rectangle de pied de page
RecreationTitleColor	H1	Couleur de titre pour récréation, énigmes
RecreationItemColor	H1	Couleur des puces dans les récréations
RecreationPartieColor	J1	Couleur des parties dans les récréation
Annexes générales		
AnnexePartieColor	A1	Couleur des parties dans les annexes
Liste des méthodes		
ListeMethodeHeadFrameColor	G3	Couleur des barres de marge dans la liste des méthodes
ListeMethodeColumnRuleColor	A1	Couleur du séparateur de colonnes dans la liste des méthodes



ListeMethodeTitleColor	B2	Couleur des titres des méthodes dans la liste des méthodes
Solutions des exercices		
CorrigeHeadFrameColor	J2	Couleur des barres de marge dans la correction des exercices
CorrigeColumnRuleColor	J1	Couleur du séparateur de colonnes dans la correction des exercices
CorrigeAETitleColor	F1	Couleur du titre « Auto-évaluation » dans la correction des exercices
CorrigeExercicesBaseTitleColor	C2	Couleur du titre « S'entraîner » dans la correction des exercices
CorrigeExercicesApprTitleColor	C2	Couleur du titre « Approfondir » dans la correction des exercices
CorrigeConnaissancesTitleColor	G1	Couleur du titre « Je teste mes connaissances » dans la correction des exercices
CorrigeQCMItemNumColor	G1	Couleur des numéros d'items vrais dans les correction de QCM
Liste des propriétés		
TablePropertyBkgColor	J4	Couleur de fond pour la colonne du milieu dans la liste des propriétés
TablePropertyTitleBkgColor	PartieStatistique	Couleur de fond des titres dans la liste des propriétés
TablePropertyTitleTextColor	Blanc	Couleur des titres dans la liste des propriétés
Lexique		
LexiqueHeadFrameColor	F2	Couleur des barres de marges dans le lexique
LexiqueColumnRuleColor	B2	Couleur du séparateur de colonnes dans le lexique
FirstLetterTxtColor	B1	Couleur de la lettre de début de groupe dans le lexique
FirstLetterBkgColor	J3	Couleur de fond de la lettre de début de groupe dans le lexique
LexiqueEntreeColor	B2	Couleur des entrées du lexique
LexiqueItemColor	B2	Couleur des puces dans le lexique

TABLE 2.2: Couleurs utilisées dans l'ouvrage

* Les couleurs définies avec * sont en fait des couleurs qui sont modifiées à chaque fois qu'on change de thème.

** Les en-têtes de page de la partie principale sont des cadres tels que :



Ces cadres utilisent six couleurs :

Couleur 0
Titre





B. Longueurs

Comme pour les couleurs, la classe offre la possibilité de modifier les longueurs. Là encore, cela va modifier la maquette initiale. La table 2.3 pages 17 à 21 liste la totalité des longueurs utilisées par la classe.

Même s'il s'agit bien de longueurs, toutes ces données sont définies en tant que macro. La seule exception est `\ActiviteTitleTopHeight` qui est réellement une longueur. Cela signifie que si toutes les longueurs de la classe peuvent être modifiées à l'aide d'une commande `\renewcommand`, pour `\ActiviteTitleTopHeight`, il faudra utiliser la commande `\setlength`.

Longueurs dans tout le document	
<code>\SquareWidth</code>	3mm
<code>\ItemRuleWidth</code>	1.5mm
<code>\ItemRuleHeight</code>	1.5mm
<code>\ItemRuleDepth</code>	0mm
<code>\ListLabelWidth</code>	4mm
<code>\EnumerateLabelWidth</code>	5mm
<code>\ListHSep</code>	2mm
<code>\AlgorithmeRuleWidth</code>	1pt
<code>\AlgorithmeSep</code>	2mm
<code>\AlgorithmeNumWidth</code>	6mm
<code>\HeadFrameWidth</code>	11cm
<code>\RoundHeadFrameWidth</code>	12.8cm
<code>\HeadTitleSep</code>	1cm
<code>\HeadTitleMaxWidth</code>	$\text{\HeadFrameWidth} - \text{\SquareWidth} * 3 - \text{\HeadTitleSep}$
<code>\RoundHeadTitleMaxWidth</code>	$\text{\RoundHeadFrameWidth} - \text{\SquareWidth} * 3 - \text{\HeadTitleSep}$
<code>\HeadFrameHeight</code>	$\text{\smc@margintop} - \text{\smc@headsep}$
<code>\HeadFrameDepth</code>	3mm
<code>\FootFrameWidth</code>	6mm
<code>\FootFrameSep</code>	1mm
<code>\AlgoIndent</code>	1em

Longueurs pour les différents cadres	
<code>\CadreSep</code>	2mm
<code>\CadreLineWidth</code>	0.5mm
<code>\FrameSep</code>	3mm
<code>\FrameLineWidth</code>	1pt
<code>\FrameArc</code>	$\text{\SquareWidth} / 2$
<code>\RuleWidth</code>	1pt
<code>\BeforeCadreVSpace</code>	1.5mm
<code>\AfterCadreVSpace</code>	1.5mm
<code>\BeforeTableVSpace</code>	1.5mm
<code>\AfterTableVSpace</code>	1.5mm
<code>\HabillageGap</code>	2mm
<code>\LogoLineWidth</code>	1pt
<code>\BeforeAlgorithmeVSpace</code>	1.5mm
<code>\AfterAlgorithmeVSpace</code>	1.5mm



Longueurs pour la page de titre

<code>\ChapterTopFrameHeight</code>	<code>2.1cm+\smc@bleed</code>
<code>\ChapterBottomFrameHeight</code>	<code>6cm</code>
<code>\ChapterTotalFrameHeight</code>	<code>\ChapterTopFrameHeight+\ChapterBottomFrameHeight</code>
<code>\ChapterBottomPicture</code>	<code>5.5cm</code>
<code>\ChapterNumLeftFrame</code>	<code>13.4cm</code>
<code>\ChapterTitleThemaWidth</code>	<code>\linewidth-\ChapterNumLeftFrame-6mm</code>
<code>\ChapterNumBase</code>	<code>3.6cm</code>
<code>\ChapterNumBottomFrame</code>	<code>2.1cm</code>
<code>\ChapterNumTopFrame</code>	<code>8.1cm+\smc@bleed</code>
<code>\ChapterTitleBase</code>	<code>2.1cm</code>
<code>\ChapterTitleSep</code>	<code>1cm</code>
<code>\ThemaBottomFrame</code>	<code>5.7cm</code>
<code>\ThemaTopFrame</code>	<code>6.6cm</code>
<code>\PrerequisTitleVSpace</code>	<code>1mm</code>
<code>\AETitleWidth</code>	<code>8cm</code>
<code>\AETitleHeight</code>	<code>\SquareWidth*3</code>
<code>\AEFrameVSep</code>	<code>4mm</code>
<code>\AEFrameHSep</code>	<code>\SquareWidth</code>
<code>\AEFrameRuleWidth</code>	<code>\SquareWidth/2</code>
<code>\AEExoFrameWidth</code>	<code>\SquareWidth*2</code>
<code>\AEExoFrameTopSep</code>	<code>1mm</code>
<code>\AfterAEExoFrameHSpace</code>	<code>2mm</code>
<code>\AECartoucheCorrSep</code>	<code>4mm</code>
<code>\AECartoucheCorrHeight</code>	<code>6mm</code>
<code>\AECartoucheCorrWidth</code>	<code>50mm</code>
<code>\AEManuelWidth</code>	<code>55mm</code>
<code>\AEManuelRightSpace</code>	<code>1mm</code>
<code>\AELogoManuelWidth</code>	<code>9mm</code>
<code>\AfterAEManuelVSpace</code>	<code>3mm</code>

Longueurs pour la partie Activités d'approche

<code>\BeforeActiviteVSpace</code>	<code>5mm</code>
<code>\ActiviteTitleTopHeight</code>	<code>6mm</code>
<code>\ActiviteTitleBottomHeight</code>	<code>1.5mm</code>
<code>\ActiviteTitleLeftSpace</code>	<code>2mm</code>
<code>\ActiviteTitleRightSpace</code>	<code>1mm</code>
<code>\AfterActiviteTitleHSpace</code>	<code>2mm</code>
<code>\AfterActiviteTitleVSpace</code>	<code>3mm</code>
<code>\BeforeActivitePartieVSpace</code>	<code>1.5mm</code>
<code>\AfterActivitePartieVSpace</code>	<code>1mm</code>

Longueurs pour la partie Cours - Méthodes

<code>\BeforeSectionVSpace</code>	<code>10mm</code>
<code>\AfterSectionVSpace</code>	<code>2mm</code>



<code>\SectionNumTitleHSpace</code>	3mm
<code>\SectionCorrectionChassePoint</code>	-0.25pt
<code>\BeforeSubsectionVSpace</code>	4mm
<code>\AfterSubsectionVSpace</code>	1mm
<code>\BeforeDefVSpace</code>	3mm
<code>\AfterDefVSpace</code>	1.5mm
<code>\longueurFrameHSep</code>	3mm
<code>\longueurFrameVSep</code>	1mm
<code>\longueurTitleFrameVSep</code>	1mm
<code>\BeforeRemVSpace</code>	3mm
<code>\AfterRemVSpace</code>	1.5mm
<code>\BeforeExempleVSpace</code>	4mm
<code>\ExempleVRuleWidthFrame</code>	1.25mm
<code>\ExempleHRuleWidthFrame</code>	0.75mm
<code>\ExempleWidthFrame</code>	15mm
<code>\ExempleHeightFrame</code>	3mm
<code>\BeforeMethodeVSpace</code>	3mm
<code>\MethodeTitleLeftSpace</code>	<code>\SquareWidth</code>
<code>\MethodeTitleRightSpace</code>	<code>\SquareWidth</code>
<code>\MethodeSubtitleLeftSpace</code>	2mm
<code>\AfterMethodeTitleVspace</code>	1.5mm
<code>\MethodeFrameHSep</code>	3mm
<code>\MethodeFrameTopSep</code>	2mm
<code>\MethodeFrameBottomSep</code>	1mm
<code>\ExAppWidthFrame</code>	30mm
<code>\ExAppHeightFrame</code>	3mm
<code>\ExAppDepthFrame</code>	0.75mm
<code>\ExAppCorrWidthFrame</code>	15mm
<code>\ExAppCorrHeightFrame</code>	3mm
<code>\ExAppCorrDepthFrame</code>	0.75mm
<code>\AfterCartoucheHSpace</code>	1.5mm
<code>\AfterExAppMethodeVSpace</code>	3mm
<code>\BeforeProofVSpace</code>	3mm
<code>\AfterProofVSpace</code>	1.5mm
<code>\ProofHeightFrame</code>	3mm
<code>\ProofDepthFrame</code>	0.75mm
<code>\ProofLeftSpace</code>	1mm
<code>\ProofRightSpace</code>	1.5mm
<code>\AfterProofTitleHSpace</code>	1.5mm

Longueurs pour les parties Exercices de base et Exercices d'approfondissement

<code>\BeforeExercicesTitleVSpace</code>	3mm
<code>\AfterExercicesTitleVSpace</code>	1.5mm
<code>\ExercicesTitleHSep</code>	3mm





<code>\ExercicesTitleVSpace</code>	2mm
<code>\BeforeExerciceVSpace</code>	1.5mm
<code>\ExerciceNumFrameWidth</code>	6mm
<code>\ExerciceNumFrameHeight</code>	3.25mm
<code>\ExerciceNumFrameDepth</code>	0.75mm
<code>\ExerciceTextHSep</code>	2mm
<code>\BeforeExercicePartieVSpace</code>	1.5mm
<code>\AfterExercicePartieVSpace</code>	1mm

Longueurs pour la partie Je teste mes connaissances

<code>\AcquisTitleVSpace</code>	1mm
<code>\BeforeQCMAEVSpace</code>	6mm plus2mm minus2mm
<code>\QCMAETitleHeight</code>	9mm
<code>\QCMAETitleWidth</code>	110mm
<code>\QCMAETitleLeftSpace</code>	17mm
<code>\QCMAEManuelWidth</code>	40mm
<code>\QCMAEManuelRightSpace</code>	1mm
<code>\QCMAELogoManuelWidth</code>	9mm
<code>\AfterQCMAEManuelVSpace</code>	0mm
<code>\AfterQCMAETitleVSpace</code>	5mm plus2mm minus1mm
<code>\AfterQCMAETextVSpace</code>	3mm plus1mm minus1mm
<code>\QCMRefHSep</code>	1pt
<code>\QCMExoFrameWidth</code>	<code>\SquareWidth*2</code>
<code>\QCMExoFrameTopSep</code>	1mm
<code>\AfterQCMExoFrameHSpace</code>	2mm
<code>\QCMLabelWidth</code>	6mm
<code>\BeforeQCMVSpace</code>	2mm plus1mm minus1mm
<code>\AfterQCMVSpace</code>	0mm plus1mm
<code>\QCMLineWidth</code>	2mm
<code>\QCMFrameSep</code>	1mm
<code>\QCMLineArc</code>	<code>\SquareWidth</code>

Longueurs pour la partie TP

<code>\BeforeTPPartieVSpace</code>	3mm plus1mm minus1mm
<code>\AfterTPPartieVSpace</code>	0mm plus1mm
<code>\TPPartieHSep</code>	2mm
<code>\BeforeRecreationVSpace</code>	6mm
<code>\AfterRecreationVSpace</code>	3mm
<code>\BeforeRecreationPartieVSpace</code>	1.5mm minus 1mm
<code>\AfterRecreationPartieVSpace</code>	1mm minus 0.5mm

Longueurs pour la liste des méthodes

<code>\ListeMethodeRuleWidth</code>	0.2pt
<code>\AfterMethodeVSpace</code>	3pt
<code>\TriangleMethodeSize</code>	1ex

Longueurs pour la partie corrigés (annexe)





<code>\CorrigeRuleWidth</code>	0.2pt
<code>\CorrigeHeadFrameWidth</code>	9mm
<code>\CorrigeChapterFrameHeight</code>	5mm
<code>\BeforeCorrigePartieTitleVSpace</code>	1mm minus 0.5mm
<code>\BeforeCorrigeChapterVspace</code>	3mm plus1mm minus1mm
<code>\QCMHSep</code>	1mm
Longueurs pour les annexes générales	
<code>\BeforeAnnexeTitleVSpace</code>	-20mm
<code>\AfterAnnexeTitleVSpace</code>	0mm plus0.5mm
<code>\AnnexeSectionRuleWidth</code>	<code>\SquareWidth</code>
<code>\AnnexeSectionRuleHSpace</code>	1mm
<code>\BeforeAnnexeSectionVSpace</code>	5mm
<code>\AfterAnnexeSectionVSpace</code>	2mm
<code>\BeforeAnnexePartieVSpace</code>	1.5mm
<code>\AfterAnnexePartieVSpace</code>	1mm
Longueurs pour le lexique	
<code>\LexiqueRuleWidth</code>	0.2pt
<code>\AfterFirstLetterVSpace</code>	2mm
<code>\BeforeFirstLetterVSpace</code>	2mm
<code>\FirstLetterFrameHSep</code>	1mm
<code>\AfterEntreeVSpace</code>	5mm

TABLE 2.3: Longueurs utilisées dans l’ouvrage

C. Commandes de fontes

La table 2.4 pages 21 à 23 montre l’ensemble des commandes de fontes effectuées par la classe. Là encore, on peut modifier ce comportement mais on ne respecte plus la maquette initiale.

Les couleurs ne sont pas intégrées dans les commandes de fontes.

Fontes dans tout le document	
<code>\AlgoFont</code>	<code>\fontsize{9.5}{9.5}\sffamily\bfseries</code>
<code>\TiceFont</code>	<code>\fontsize{9.5}{9.5}\sffamily\bfseries</code>
Fontes de la page de titre	
<code>\ThemaTitleFont</code>	<code>\fontsize{20}{24}\sffamily</code>
<code>\ChapterTitleFont</code>	<code>\fontsize{40}{40}\sffamily\bfseries</code>
<code>\ChapAppFont</code>	<code>\fontsize{15}{18}\sffamily</code>
<code>\ChapterNumFont</code>	<code>\fontsize{88}{106}\sffamily</code>
<code>\PrerequisTitleFont</code>	<code>\fontsize{14.4}{17.28}\sffamily\bfseries</code>
<code>\PrerequisTextFont</code>	<code>\fontsize{9.5}{14.4}\sffamily</code>
<code>\AETitleFont</code>	<code>\fontsize{15}{18}\sffamily\bfseries</code>
<code>\AEExoNumFont</code>	<code>\fontsize{9.5}{14.4}\sffamily\bfseries</code>
<code>\AECartoucheCorrArrowFont</code>	<code>\fontsize{12}{14.4}</code>



\AECartoucheCorrVCPFont	\fontsize{9.5}{14.4}\sffamily
\AEManuelFont	\fontsize{9.5}{10.5}\sffamily
\LogoManuelFont	\fontsize{9.5}{10.5}\sffamily\bfseries
\LogoURLManuelFont	\fontsize{9.5}{10.5}\sffamily\bfseries
Fontes pour les en-têtes et pied de page	
\HeadTitleFont	\fontsize{24}{28.8}\sffamily\bfseries
\RoundHeadTitleFont	\fontsize{24}{28.8}\sffamily\bfseries
\FootChapterNumFont	\fontsize{12}{14.4}\sffamily\bfseries
\FootTitleHeadFont	\fontsize{9.5}{12}\sffamily\bfseries
\FootTitleChapterFont	\fontsize{9.5}{12}\sffamily
\FootPageFont	\fontsize{9.5}{12}\sffamily\bfseries
Fontes pour la partie Activités d'approche	
\ActiviteTitleFont	\fontsize{15}{18}\sffamily\bfseries
\ActiviteNumFont	\fontsize{15}{18}\sffamily\bfseries
\ActiviteSubtitleFont	\fontsize{15}{18}\sffamily\bfseries
\ActivitePartieFont	\fontsize{12}{14.4}\sffamily\bfseries
Fontes pour la partie Cours - Méthodes	
\SectionNumFont	\fontsize{21}{25}\sffamily\bfseries
\SectionTitleFont	\fontsize{17.28}{21}\sffamily\bfseries
\SubsectionNumFont	\fontsize{17.28}{21}\sffamily\bfseries
\SubsectionTitleFont	\fontsize{14.4}{17.28}\sffamily\bfseries
\DefTitleFont	\fontsize{12}{14.4}\sffamily
\DefSubtitleFont	\fontsize{12}{14.4}\sffamily\bfseries
\RemTitleFont	\fontsize{9.5}{14.4}\scshape\bfseries
\ExempleTitleFont	\fontsize{9.5}{14.4}\sffamily
\MethodeTitleFont	\fontsize{12}{14.4}\sffamily\bfseries
\MethodeSubtitleFont	\fontsize{12}{14.4}\sffamily\bfseries
\MethodeRefExerciceFont	\fontsize{9.5}{14.4}\sffamily
\ExAppTitleFont	\normalsize\sffamily
\ExAppCorrTitleFont	\normalsize\sffamily
\ProofTitleFont	\fontsize{9.5}{14.4}\sffamily
Fontes pour les parties Exercices de base et Exercices d'approfondissement	
\ExercicesTitleFont	\fontsize{14.4}{17.28}\sffamily\bfseries
\ExerciceNumFont	\fontsize{10.5}{14.4}\sffamily\bfseries
\ExerciceTitleFont	\fontsize{10.5}{14.4}\sffamily\bfseries
\ExercicePartieFont	\fontsize{10.5}{14.4}\bfseries
\ExerciceRefMethodeFont	\fontsize{9.5}{14.4}\sffamily\bfseries
\ExercicePageRefMethodeFont	\fontsize{9.5}{14.4}\sffamily
Fontes pour la partie Je teste mes connaissances	
\AcquisTitleFont	\fontsize{17.28}{17.28}\sffamily\bfseries
\AcquisTextFont	\fontsize{9.5}{14.4}\sffamily
\QCMATitleFont	\fontsize{20}{20}\sffamily\bfseries
\QCMAManuelFont	\fontsize{9.5}{10.5}\sffamily



<code>\QCMATextAfterTitleFont</code>	<code>\bfseries</code>
<code>\QCMExoNumFont</code>	<code>\fontsize{9.5}{14.4}\sffamily\bfseries</code>
<code>\QCMItemNumFont</code>	<code>\fontsize{9.5}{14.4}\sffamily\bfseries</code>
Fontes pour la partie TP	
<code>\TPPartieFont</code>	<code>\fontsize{12}{14.4}\sffamily\bfseries</code>
<code>\RecreationFont</code>	<code>\fontsize{20}{20}\bfseries</code>
<code>\RecreationTitleFont</code>	<code>\fontsize{12}{14.4}\sffamily\bfseries</code>
<code>\RecreationPartieFont</code>	<code>\fontsize{10}{14.4}\BFSERIES</code>
<code>\RecreationPartieTitleFont</code>	<code>\fontsize{10}{14.4}\bfseries</code>
Fontes pour la liste des méthodes	
<code>\ListeMethodeChapterFont</code>	<code>\fontsize{40}{40}\sffamily</code>
<code>\ListeMethodeTitleFont</code>	<code>\fontsize{12}{14.4}\sffamily\bfseries</code>
<code>\ListeMethodePageFont</code>	<code>\fontsize{12}{14.4}\sffamily\bfseries</code>
Fontes pour la partie Corrigés (annexe)	
<code>\CorrigeTitleFont</code>	<code>\fontsize{40}{40}\sffamily</code>
<code>\CorrigeChapterFont</code>	<code>\fontsize{12}{14.4}\sffamily\bfseries</code>
<code>\CorrigePartieFont</code>	<code>\fontsize{11}{14.4}\sffamily\bfseries</code>
<code>\CorrigeNumExerciceFont</code>	<code>\fontsize{9.5}{14.4}\sffamily\bfseries</code>
<code>\CorrigeQCMItemNumFont</code>	<code>\fontsize{9.5}{14.4}\sffamily</code>
Fontes pour les annexes générales	
<code>\FootAnnexeTxtFont</code>	<code>\fontsize{11}{14.4}\sffamily</code>
<code>\FootAnnexePageFont</code>	<code>\fontsize{11}{14.4}\sffamily\bfseries</code>
<code>\AnnexeTitleFont</code>	<code>\fontsize{35}{35}\sffamily</code>
<code>\AnnexeSectionTitleFont</code>	<code>\fontsize{14.4}{14.4}\sffamily\bfseries</code>
<code>\ProprieteFont</code>	<code>\fontsize{10}{14.4}\sffamily</code>
<code>\NumProprieteFont</code>	<code>\fontsize{10}{14.4}\sffamily\bfseries</code>
<code>\AnnexePartieFont</code>	<code>\fontsize{10.5}{14.4}\bfseries</code>
Fontes pour le lexique	
<code>\LexiqueTitleFont</code>	<code>\fontsize{50}{50}\sffamily</code>
<code>\FirstLetterFont</code>	<code>\fontsize{16}{16}\sffamily\bfseries</code>
<code>\LexiqueEntreeFont</code>	<code>\fontsize{12}{14.4}\sffamily\bfseries</code>
<code>\LexiqueFont</code>	<code>\sffamily\bfseries</code>

TABLE 2.4: Fontes utilisées dans l'ouvrage

D. Textes fixes

La classe définit tous les textes fixes dans des macros. Cela rend très facile la modification de ces textes mais, encore une fois, on ne respecte alors plus la maquette initiale. La table 2.5 indique la liste de tous ces textes fixes.

<code>\StringColitemize</code>	<code>colitemize</code>
<code>\StringColenumerate</code>	<code>colenumerate</code>
<code>\StringPrerequis</code>	Connaissances du collège nécessaires à ce chapitre



<code>\StringActivitesApproche</code>	Activités d'approche
<code>\StringCoursMethodes</code>	Cours - Méthodes
<code>\StringExercicesBase</code>	S'entraîner
<code>\StringExercicesAppr</code>	Approfondir
<code>\StringTestConnaissances</code>	Je teste mes connaissances
<code>\StringConnaissances</code>	Auto-évaluation QCM
<code>\StringTravauxPratiques</code>	Travaux pratiques
<code>\StringAE</code>	Auto-évaluation
<code>\StringVoirCorriges</code>	Voir solutions p.
<code>\StringManuel</code>	Des ressources numériques pour préparer le chapitre sur <code>\textcolor{U4}{\LogoURLManuelFont manuel.sesamath.net}</code>
<code>\StringALGO</code>	ALGO
<code>\StringTICE</code>	INFO
<code>\StringRecreation</code>	Récréation, énigmes
<code>\StringAcquisTitle</code>	À la fin de ce chapitre, je dois être capable de :
<code>\StringQCMAE</code>	QCM d'auto-évaluation
<code>\StringQCM</code>	QCM
<code>\StringChapitre</code>	Chapitre
<code>\StringEnigme</code>	Énigme
<code>\StringPropriete</code>	Propriété
<code>\StringPROPRIETE</code>	PROPRIÉTÉ
<code>\StringDEFINITION</code>	DÉFINITION
<code>\StringTHEOREME</code>	THÉORÈME
<code>\StringLexique</code>	Lexique
<code>\LogoManuel</code>	<code>\textcolor{U4}{\LogoManuelFont @}</code>
<code>\StringDEBAT</code>	DÉBAT
<code>\StringACTIVITE</code>	ACTIVITÉ
<code>\StringTP</code>	TP
<code>\StringExemple</code>	Exemple
<code>\StringCorrection</code>	Correction
<code>\StringMETHODE</code>	MÉTHODE
<code>\StringPREUVE</code>	PREUVE
<code>\StringMethode</code>	Méthode
<code>\StringExApp</code>	Exercice d'application
<code>\StringPartie</code>	Partie
<code>\StringListeMethode</code>	Liste des méthodes
<code>\StringCorriges</code>	Solutions
<code>\StringListeMethodeF</code>	Fonctions
<code>\StringListeMethodeG</code>	Géométrie
<code>\StringListeMethodeSP</code>	Stat. et probabilités

TABLE 2.5: Textes fixes utilisées dans l'ouvrage





ATTENTION : Les macros `\StringColitemize`, `\StringColenumerate` et `\StringQCM` ne doivent surtout pas être modifiées sous peine d’entraîner des erreurs de compilation.

2. Première page d’un chapitre

La première page de chaque chapitre est dédiée à un rappel sur les connaissances de collège (par défaut) nécessaire pour ce chapitre et d’une série de petits exercices appelée « Auto-évaluation ».

ATTENTION : La maquette est très stricte à ce niveau : il faut que ces deux éléments tiennent sur une page et la remplissent au mieux. La classe pose un ressort vertical infini au-dessus du cadre bleu avec son titre rouge et entre ce cadre bleu et la partie Auto-évaluation. Il revient à l’auteur de remplir cet espace au mieux. De façon générale, cette maquette interdit toute coupure de page au niveau des cadres et d’un grand nombre de structures.

Le cadre bleu avec le titre en rouge est composé avec l’environnement `prerequis`. Il est destiné à indiquer quelles sont les connaissances des années antérieures que l’élève est supposé maîtriser. Comme la maquette traite de livres de seconde, le texte par défaut est « Connaissances du collège nécessaires à ce chapitre ». On peut modifier ce titre de deux façons :

SYNTAXE :

soit de façon locale avec un argument optionnel

Le code

```
\begin{prerequis}[Nouveau titre]
...
\end{prerequis}
```

soit de façon globale en modifiant la macro `\StringPrerequis`.

Le code

```
\renewcommand*\StringPrerequis{Nouveau titre}
```

Le titre est écrit en rouge au-dessus d’un cadre bleu et le contenu de l’environnement sera composé en double colonne à l’intérieur du cadre bleu. La maquette prévoit que la liste des connaissances soit exposée avec un environnement `itemize`.

REMARQUE : L’environnement `itemize` doit se situer à l’intérieur de l’environnement `prerequis`. La classe interdit l’utilisation des listes `itemize` ou `enumerate` en dehors de certains endroits.

De plus, les puces et les couleurs utilisées dépendront également de l’endroit où on utilise ces listes. Pour les couleurs, on se référera au tableau 2.2. Pour l’aspect, on dresse la liste de toutes les listes possibles aux pages 53-54.

À la suite des prérequis, on trouve des exercices d’auto-évaluation. Ceux-ci sont composés dans l’environnement `autoeval`. Chaque exercice est composé dans un environnement `exercice`. La correction sera indiquée immédiatement après dans un environnement `corrige`. Par exemple, les exercices de la première page de ce chapitre ont été composés de la façon suivante :



SYNTAXE :

```
\begin{autoeval}
  \begin{multicols}{2}
    \begin{exercice}
      Serez-vous capable de lire ce manuel en entier en moins de
      deux heures ?
    \end{exercice}
    \begin{corrige}
      Soit vous disposez d'une capacité hors-norme pour la vitesse de
      lecture, soit il ne faut pas trop se faire d'illusion et il est
      impossible d'assimiler l'ensemble des fonctionnalités en moins
      de deux heures.
    \end{corrige}
    ...
  \end{multicols}
\end{autoeval}
```

et on peut effectivement voir page 74 le texte de la solution au premier exercice.

Une présentation plus en détail des solutions des exercices sera donnée à la section 12 page 41.

3. Activités d'approche

La partie Activités d'approche est censée être une succession de débats et d'activités. Les débats sont définis par l'environnement `debat` et les activités par l'environnement `activite`.

La syntaxe est la même pour ces deux environnements, à savoir la possibilité d'indiquer deux arguments optionnels. Le premier argument indique le titre du débat ou de l'activité et le deuxième argument optionnel indique la présence d'un logo à droite du titre. Si les arguments optionnels ne sont pas spécifiés, l'élément correspondant n'existera pas (pas de titre ou pas de logo).

La classe ne prévoit que deux logos : le logo pour les algorithmes avec la commande `\algo` qui donne le résultat **ALGO** et le logo pour les tice avec la commande `\tice` qui donne le résultat **INFO**. On peut trouver l'offre un peu limitée et vouloir inventer d'autre logo. Si le logo doit uniquement être un texte dans un cartouche, autant être homogène dans la présentation et se servir de la trame générale de la classe. Celle-ci définit `\algo` de la façon suivante :

```
\newcommand*\algo{%
  \psframebox[framesep=1pt,linewidth=\LogoLineWidth,
             linecolor=AlgoLineColor, fillstyle=solid,
             fillcolor=AlgoBkgColor, framearc=0.6]{%
    \AlgoFont
    \textcolor{AlgoTextColor}{\StringALGO}%
  }
}
```



La commande `\AlgoFont` vaut `\fontsize{9.5}{9.5}\sffamily\bfseries` comme l'indique le tableau 2.4 pages 21 à 23. Les définitions des couleurs sont indiquées dans le tableau 2.2 pages 10 à 16. Les éléments les plus susceptibles d'être modifiés sont les couleurs et, évidemment, le texte à l'intérieur du cartouche.

Voici un exemple de syntaxe avec un titre et un logo.

Le code

```
\begin{activite}[Titre de l'activité][\tice]
  Texte de l'activité...
\end{activite}
\begin{debat}[Titre du débat][\algo]
  Texte du débat...
\end{debat}
```

donne le résultat :

ACTIVITÉ 1 Titre de l'activité

INFO

Texte de l'activité...

DÉBAT 2 Titre du débat

ALGO

Texte du débat...

En fait, ces environnements sont définies par la classe en utilisant la commande `\DeclareActivityLike`. Cette commande demande cinq arguments qui sont :

- 1) le nom de l'environnement
- 2) Le titre de l'environnement (dans le cartouche)
- 3) La couleur du fond de cartouche
- 4) La couleur du trait sous le cartouche
- 5) La couleur du titre de l'environnement (et du losange du numéro)

Ainsi, on a le code suivant dans la classe :

```
\DeclareActivityLike{debat}{\StringDEBAT}
  {ActiviteDebatTopColor}
  {ActiviteDebatBottomColor}
  {ActiviteTitleColor}
\DeclareActivityLike{activite}{\StringACTIVITE}
  {ActiviteActiviteTopColor}
  {ActiviteActiviteBottomColor}
  {ActiviteTitleColor}
```



MÉTHODE 2 Créer un nouvel environnement dans les pages Activités

Exercice d'application

Le code

```
\DeclareActivityLike{nouveau}{DÉCOUVERTE}{lime}{orange}{red}
\begin{nouveau}
  Un nouvel environnement de type activité
\end{nouveau}
```

Correction

donne le résultat :

DÉCOUVERTE 3

Un nouvel environnement de type activité

À l'intérieur des activités et des débats, on peut utiliser les listes et les parties, on ne peut pas définir d'exercice.

4. Cours-Méthodes

La partie cours et méthodes est activée avec la commande `\cours`.

Sectionnement

Cette partie est composée de sections avec des sous-sections éventuelles.

Les sections et les sous-sections utilisent les commandes standards `\section` et `\subsection` avec leur syntaxe standard. On a même le droit à la version étoilée de ces commandes bien que les versions étoilées et non étoilées soient strictement équivalentes pour la classe `sesamanuel`. De même, il y a possibilité d'indiquer un argument optionnel mais celui-ci n'a aucun rôle... du moins dans la version actuelle de la classe !

La classe interdit toutes les autres commandes de sectionnement. Ainsi les commandes `\part`, `\subsubsection`, `\paragraph` et `\subparagraph` donneront une erreur de compilation.

Cette documentation utilise essentiellement la partie Cours-Méthodes et les sections y sont utilisées.

Pour les sous-sections, le code

Le code

```
\subsection{Titre de sous-section}
```

donne le résultat :

A. Titre de sous-section



■ DÉFINITION : Titre optionnel

Dans le cours, on utilise assez souvent des cadres du type définition (comme ici par exemple).

Le code

```
\begin{definition}[Titre optionnel]
Dans le cours, on utilise assez souvent des cadres du type
définition (comme ici par exemple).
\end{definition}
```

génère la définition ci-dessus.

REMARQUE : On peut voir que le titre de la définition est spécifiée avec un argument optionnel. Si celui-ci n'existe pas, il n'y a ni titre, ni deux points après le mot « DÉFINITION ». En réalité, la classe utilise la commande `\DeclareDefLike` pour définir plusieurs environnements de ce type. Cette commande demande deux arguments : le premier est le nom de l'environnement et le deuxième est le mot qui sera utilisé comme titre principal. La classe définit ainsi trois environnements avec le code :

```
\DeclareDefLike{definition}{\StringDEFINITION}
\DeclareDefLike{propriete}{\StringPROPRIETE}
\DeclareDefLike{theoreme}{\StringTHEOREME}
```

Il est donc facile de se créer d'autres environnements de ce type en cas de besoin.

■ Remarque

À côté des environnements de type définition, on a des environnements de type remarque.

Le code

```
\begin{remarque}
Ceci est une remarque.
\end{remarque}
```

donne le résultat :

REMARQUE : Ceci est une remarque.

Comme pour les environnements de type définition, les environnements de type remarque sont définis en utilisant une commande générale, en l'occurrence la commande `\DeclareRemLike`. Cette commande ne demande que deux paramètres : le nom de l'environnement et le titre de l'environnement. La classe utilise cette commande pour définir un certain nombre d'environnement avec le code :

```
\DeclareRemLike{remarque}{Remarque}
\DeclareRemLike{remarques}{Remarques}
\DeclareRemLike{notation}{Notation}
\DeclareRemLike{notations}{Notations}
\DeclareRemLike{vocabulaire}{Vocabulaire}
```



■ Preuve

Le dernier type d'environnement de ce type est l'environnement preuve qui sert à introduire les démonstrations. Sa syntaxe est aussi simple que celle des environnements de type remarque.

▣ Le code

```
\begin{preuve}
  Ceci est une preuve\par Deuxième ligne de la preuve
\end{preuve}
```

donne le résultat

▣ **PREUVE** Ceci est une preuve
Deuxième ligne de la preuve

Comme pour les autres environnements, cet environnement est créé en utilisant une commande. Ici, c'est la commande `\DeclareProofLike`. Elle demande deux arguments qui sont le nom de l'environnement et son titre. La classe l'utilise uniquement pour définir l'environnement preuve avec le code

```
\DeclareProofLike{preuve}{\StringPREUVE}
```

Elle a été utilisé pour créer un environnement code pour mettre en valeur les exemples de code dans cette documentation.

■ Exemple

On utilise aussi deux environnements permettant d'afficher un exemple avec sa correction associée. Le premier compose l'exemple et la correction en vis-à-vis et le second compose l'exemple et la correction l'une au-dessous de l'autre. La correction est introduite avec la macro `\correction`. L'environnement qui compose en vis-à-vis est `exemple` et celui qui compose verticalement est `exemple*1`.

▣ Le code

```
\begin{exemple}
  Texte de l'exemple
\correction
  Texte de la correction
\end{exemple}
\begin{exemple*1}
  Texte de l'exemple
\correction
  Texte de la correction
\end{exemple*1}
```

donne le résultat

▣ Exemple

Texte de l'exemple

▣ Correction

Texte de la correction

▣ Exemple Texte de l'exemple

▣ Correction Texte de la correction

Le premier environnement accepte un argument optionnel pour indiquer le rapport de l'empagement à disposition affecté au texte de l'exemple. Par défaut, ce rapport vaut 0.4. En jouant avec ce rapport, on peut équilibrer un peu plus la hauteur des deux colonnes dans certaines circonstances.





■ Méthode

La dernière structure complexe rencontrée dans la partie Cours et Méthode est celle consacrée aux méthodes. La classe `sesamaneu1` propose une gestion très fine de cet élément. Les méthodes présentent :

- un titre (optionnel mais présent la plupart du temps) ;
- un texte introductif (optionnel) ;
- un ou deux exercices d'application ;
- une ou deux corrections correspondantes ;

Il y a quatre types possibles de présentation représentés par quatre environnements différents :

- L'environnement `methode` qui compose la méthode avec l'exercice d'application en vis-à-vis de sa correction.
- L'environnement `methode*1` qui compose l'ensemble de la méthode sur une seule colonne de la largeur d'empagement.
- L'environnement `methode*2` qui compose une méthode avec l'exercice d'application et sa correction sur deux colonnes (mais pas nécessairement en vis-à-vis).
- L'environnement `methode*2*2` qui compose une méthode avec deux colonnes contenant chacune un exercice d'application et sa correction au-dessous.

Le code

```
\begin{methode}[Titre de la méthode]
  Texte introductif
  \exercice
  Texte de l'exercice
  \correction
  Texte de la correction sur un minimum de trois lignes pour faire la
  différence entre vis-à-vis et double colonne. C'est l'endroit de la
  coupure qui va différer.
\end{methode}
```

donne le résultat :

MÉTHODE 3 Titre de la méthode

Texte introductif

Exercice d'application

Texte de l'exercice

Correction

Texte de la correction sur un minimum de trois lignes pour faire la différence entre vis-à-vis et double colonne. C'est l'endroit de la coupure qui va différer.



Le code

```
\begin{methode*1}[Titre de la méthode]
  [même corps que ci-dessus]
\end{methode*1}
```

donne le résultat :

MÉTHODE 4 Titre de la méthode

Texte introductif

Exercice d'application Texte de l'exercice

Correction Texte de la correction sur un minimum de trois lignes pour faire la différence entre vis-à-vis et double colonne. C'est l'endroit de la coupure qui va différer.

Le code

```
\begin{methode*2}[Titre de la méthode]
  [même corps que ci-dessus]
\end{methode*2}
```

donne le résultat :

MÉTHODE 5 Titre de la méthode

Texte introductif

Exercice d'application Texte de l'exercice

Correction Texte de la correction sur un minimum

de trois lignes pour faire la différence entre vis-à-vis et double colonne. C'est l'endroit de la coupure qui va différer.

et on voit bien la différence entre les environnements `methode` et `methode*2` : le premier composant en vis-à-vis alors que le second fait une composition sur deux colonnes en équilibrant celles-ci.



Le dernier environnement de type « méthode » permet de composer deux exercices avec les deux corrections le tout sur deux colonnes.

Le code

```
\begin{methode*2*2}[Dernière méthode\MethodeRefExercice{exo-exemple}]
  \exercice
  \label{methode-exemple}
  Texte du premier exercice
  \correction
  Correction du premier exercice
  \exercice
  Texte du deuxième exercice
  \correction
  Texte de la correction du deuxième exercice sur un minimum de trois
  lignes pour faire la différence entre vis-à-vis et double
  colonne. C'est l'endroit de la coupure qui va différer.
\end{methode*2*2}
```

donne le résultat :

MÉTHODE 6 Dernière méthode

► Ex. 3 p. 34

Exercice d'application Texte du premier exercice

Correction Correction du premier exercice

Exercice d'application Texte du deuxième exercice

Correction Texte de la correction du deuxième exercice sur un minimum de trois lignes pour faire la différence entre vis-à-vis et double colonne. C'est l'endroit de la coupure qui va différer.

REMARQUE : Il est possible qu'une méthode soit directement reliée à un exercice qui apparaîtra plus loin (dans la partie Exercices de base ou Exercices d'approfondissement). Dans ce cas, on peut indiquer une référence à cet exercice à l'intérieur du titre de la méthode avec la macro `\MethodeRefExercice` qui va prendre en argument l'étiquette indiquée (avec un `\label`) au niveau de l'exercice en question. De plus, il est fortement recommandé que l'exercice fasse référence à la méthode avec la macro `\ExerciceRefMethode` qui prendra en argument l'étiquette indiquée au niveau de la méthode (toujours avec la commande `\label`). On reverra ce point dans la section suivante avec un exemple complet. La macro `\MethodeRefExercice` peut être étoilée mais là aussi, on reverra cela avec les exercices.



5. Exercices d'entraînement

Les exercices sont répartis en exercices d'entraînement et en exercices d'approfondissement.

La partie Exercices d'entraînement est introduite par la macro `\exercicesbase`. La maquette demande à ce que les exercices soient composés en double-colonne avec un filet de séparation d'une certaine épaisseur et d'une certaine couleur. Tout ceci est réalisé en plaçant les exercices dans l'environnement `colonne*exercice`.

On peut regrouper les exercices par sous-thèmes avec la commande `\serie` qui demande un argument obligatoire : le thème des exercices.

Les exercices proprement dits seront spécifiés à l'intérieur d'un environnement `exercice` qui a exactement la même syntaxe que pour la première page d'un chapitre, c'est-à-dire un premier argument optionnel pour un titre d'exercice et un deuxième argument optionnel pour un logo.

L'environnement étoilé est normalement destiné aux exercices corrigés. En réalité, il ne fait que donner au cadre du numéro d'exercice une autre couleur de fond, c'est la présence de l'environnement `corrige` qui fera qu'il y aura une correction. Ces corrections, ainsi que toutes les autres corrections de toutes les autres parties des chapitres seront affichés dans la partie Solutions (voir section 12 page 41).

Le code

```
\begin{colonne*exercice}
\serie{titre de série}
\begin{exercice}[Exercice sans correction][\tice]
  prouver que  $1=1$ 
\end{exercice}

\begin{exercice*}[Exercice avec correction]
  Prouver que  $2=2$ 
\end{exercice*}
\begin{corrige}
  On sait que  $1=1$  avec l'exercice précédent donc  $1+1=1+1$ ,
  c'est-à-dire  $2=2$ .
\end{corrige}

\begin{exercice}[Lien avec une méthode \ExerciceRefMethode{methode-exemple}]
  \label{exo-exemple}
  Test pour avoir un lien avec une méthode.
\end{exercice}
\end{colonne*exercice}
```

donne le résultat :

titre de série

1 Exercice sans correction

prouver que $1 = 1$

INFO

2 Exercice avec correction

Prouver que $2 = 2$

3 Lien avec une méthode

Test pour avoir un lien avec une méthode.

► MÉTHODE 6 p. 33



REMARQUE :

- L'exemple montre également le jeu de ping-pong avec les méthodes. L'exercice 3 fait référence à la méthode 6 page 33 grâce à la commande `\ExerciceRefMethode`. En revenant au code de cette méthode, on peut voir que cette dernière fait référence à cet exercice grâce à la commande `\MethodeRefExercice` (voir le code page 33).
- À l'intérieur d'un exercice, il est possible d'utiliser la macro `\partie`. Cette commande accepte un argument optionnel qui sera le titre de la partie.

6. Exercices d'approfondissement

La partie Exercices d'approfondissement est introduite par la commande `\exercicesappr`. Cela va commencer une nouvelle page et modifier le titre des en-têtes mais tout le reste sera strictement identique à la partie Exercices de base.

7. Récréation, énigmes

À la suite des exercices d'approfondissement (ou des travaux pratiques), on peut placer un ou plusieurs exercices plus délicats ou sortant de l'ordinaire qu'on regroupe dans une partie Récréation, énigmes. Contrairement aux autres parties des chapitres, cette partie peut être introduite de deux façons différentes. Soit avec la commande `\recreation`, soit avec la commande `\Recreation` : on notera la première lettre capitale ou non.

La version `\recreation` ne provoque pas de coupure de page et pas d'en-tête particulière. Un bandeau similaire à ce qu'on trouve en en-tête sera composé en milieu de page.

Le code

```
\recreation
```

donne le résultat :

Récréation, énigmes

La version `\Recreation` (avec sa première lettre capitale) provoque un changement de page et un changement d'en-tête de page (ainsi qu'un changement de couleur de pied de page) comme toutes les autres commandes de début de partie de chapitre.

Dans la partie Récréation-énigmes, on utilise l'environnement `enigme` pour spécifier les textes d'exercices. Ceux-ci ne sont pas numérotés donc cette partie est plutôt destinée à recevoir un seul exercice. C'est encore plus vrai avec la commande `\recreation` qui ne provoque pas de changement d'en-tête et qui est donc vraiment destiné à recevoir du matériel sur le reste de la page, pas sur la page suivante.



8. Je teste mes connaissances

La partie Je teste mes connaissances est introduite par la commande `\connaissances`. Comme les autres commandes de partie, elle provoque un saut de page et la modification des en-têtes et pieds de page.

■ À la fin du chapitre je dois être capable de

Dans cette partie, on attend une première zone où on indique les points devant être acquis en fin de leçon puis une seconde zone avec des QCM d'auto-évaluation.

La zone indiquant les points devant être acquis est indiquée avec l'environnement `acquis`. Le corps de cet environnement liste ces points (normalement dans un environnement `itemize` éventuellement plusieurs environnements emboîtés).

■ Le Q.C.M.

- La zone des QCM d'auto-évaluation est introduite par la commande `\QCMautoevaluation` qui demande un argument obligatoire pour un texte informatif général. C'est normalement un texte du type « Pour chaque question, plusieurs réponses sont proposées. Déterminer celles qui sont correctes. »
- À l'intérieur de cet environnement, on peut définir des cadres à l'intérieur desquels on regroupera un texte introductif et plusieurs QCM apparentés.
 - Le cadre extérieur se réalise en appelant l'environnement `QCM`, le texte introductif commun est composé avec l'environnement `EnonceCommunQCM` et les QCM seront tous placés dans un environnement `GroupeQCM`. Dans ce dernier environnement, les QCM et leurs corrigés seront saisis comme précédemment, c'est-à-dire avec les environnements `exercice` et `corrige`.
 - Les QCM sont des exercices un peu particuliers et certaines commandes sont faites pour eux. On trouve l'environnement `ChoixQCM` qui va se comporter comme une liste numérotée en colonne avec la commande `\item` pour indiquer les différentes propositions. Elle demande un argument obligatoire qui est le nombre de colonnes.
 - Pour les corrections, on trouve la commande `\reponseQCM` qui demande un argument obligatoire donnant la liste des réponses justes. Si plusieurs réponses sont justes, on donnera la liste des lettres sans séparateur.

Pour illustrer toutes ces informations :





Le code

```
\connaissances
\begin{acquis}
  \begin{itemize}
    \item Premier point à connaître.
    \item Autre point à savoir faire.
    \item Dernier point devant être su.
  \end{itemize}
\end{acquis}
\QCMautoevaluation{texte introductif}
\begin{QCM}
  \begin{EnonceCommunQCM}
    Pour les questions \RefQCM{premier-qcm} à
    \RefQCM{deuxieme-qcm},  $f$  désigne une
    fonction affine.
  \end{EnonceCommunQCM}
  \begin{GroupeQCM}
    \begin{exercice}\label{premier-qcm}
      La courbe de  $f$  est
      \begin{ChoixQCM}{3}
        \item une droite
        \item une parabole
        \item autre
      \end{ChoixQCM}
    \end{exercice}
    \begin{corrige}
      \reponseQCM{a}
    \end{corrige}
    \begin{exercice}\label{deuxieme-qcm}
       $f(3)$ 
      \begin{ChoixQCM}{3}
        \item vaut la moitié de  $f(6)$ 
        \item vaut le double de  $f(6)$ 
        \item on ne peut pas savoir
      \end{ChoixQCM}
    \end{exercice}
    \begin{corrige}
      \reponseQCM{c}
    \end{corrige}
  \end{GroupeQCM}
\end{QCM}
```

Le résultat est indiqué page suivante :



À la fin de ce chapitre, je dois être capable de :

- ▶ Premier point à connaître.
- ▶ Autre point à savoir faire.
- ▶ Dernier point devant être su.



QCM d'auto-évaluation

Des ressources numériques
pour préparer le chapitre sur
manuel.sesamath.net



texte introductif

Pour les questions 4 à 5, f désigne une fonction affine.

4 La courbe de f est

- a une droite b une parabole c autre

5 $f(3)$

- a vaut la moitié de $f(6)$ b vaut le double de $f(6)$ c on ne peut pas savoir



9. Travaux pratiques

La partie concernant les travaux pratiques est introduite avec la macro `\TravauxPratiques`.

Elle se compose uniquement de travaux pratiques. Ceux-ci sont indiqués grâce à l'environnement TP.

Il s'agit d'un environnement qui fonctionne exactement comme les activités et les débats (voir section 3 page 26).

Tout ce qui a été dit pour les activités reste valable. En particulier, la possibilité d'avoir un titre avec le premier argument optionnel de l'environnement et un logo avec le second argument optionnel. On peut même se créer de nouveaux environnements avec `\DeclareActivityLike` si on trouve que le titre « Activité » n'est pas le plus approprié. N'oubliez cependant pas que tout est paramétrable : le mot « TP » qui est en fait le contenu de la macro `\StringTP` et les couleurs utilisées qui sont `TPTopColor`, `TPBottomColor` et `TPTitleColor`.

Dans l'énoncé d'un TP, on peut utiliser la commande `\partie`.

Il est possible de terminer une série de travaux pratiques par une Récréation, énigmes (voir section 7 page 35).

10. Annexes générales

Toutes les annexes sont placées en fin d'ouvrage, à la suite des chapitres. Un certain nombre d'annexes ont des usages très particuliers et seront vues dans les sections qui suivent. On peut quand même signaler que toutes les annexes partageront le même type d'en-tête qui, en fait ne sera pas vraiment une en-tête mais une sorte de bandeau courant sur le bord de la page. Seule sa couleur changera selon le type d'annexe.

On commencera par les annexes sans caractéristique particulière qui sont introduites avec la macro `\annexe`. Cette commande demande un argument obligatoire pour le titre de l'annexe.

À l'intérieur d'une annexe, on peut presque tout faire. Les deux seuls éléments interdits sont les corrections d'exercices et la commande `\partie` (lorsqu'elle est utilisée en dehors d'un exercice).

Il y a une certaine homogénéité dans le choix des couleurs à l'intérieur d'une annexe. Il est possible, comme d'habitude, de modifier les couleurs de façon individuelle mais, pour assurer cette homogénéité la classe fournit la commande `\ChangeAnnexe` qui demande quatre arguments qui seront les couleurs utilisées dans l'annexe. Le premier argument est la couleur du bandeau de marge (`AnnexeHeadFrameColor`). Le deuxième argument est une couleur partagée par plusieurs éléments :

- `AnnexeSectionRuleColor`
- `AnnexeItemColor`
- `AnnexeExerciceCorrigeNumFrameColor`
- `AnnexeExerciceNumFrameColor`

Le troisième argument est également une couleur partagée par plusieurs éléments :

- `AnnexeSectionTitleColor`
- `AnnexeExerciceTitleColor`

Enfin, le quatrième et dernier argument est la couleur des numéros d'exercices. Par défaut la classe appelle cette macro de cette façon :

```
\ChangeAnnexe{G3}{A1}{G1}{Blanc}
```



11. Liste des méthodes

L'annexe spéciale réservée à la liste des méthodes ne demande rien d'autre à l'utilisateur qu'à taper la commande `\AfficheListeMethodes`.

SYNTAXE :

```
\AfficheListeMethodes[nombre colonnes]
```

L'argument optionnel indique le nombre de colonnes sur lesquelles composer la liste des méthodes. Par défaut, c'est la valeur de la macro `\NombreColonnesListeMethodes` et par défaut, la classe définit cette dernière comme valant 2.

Les méthodes sont listées en commençant par le thème Statistiques et probabilités, puis par le thème Fonctions et enfin par le thème Géométrie. Cet ordre peut être modifié en redéfinissant la commande `\ListeMethodesThemes`. Cette dernière est une liste, séparée par des virgules, des lettres minuscules et capitales définissant chaque thème. La définition initiale donnée par la classe est donc :

```
\newcommand\ListeMethodesThemes{{sp}{SP},{f}{F},{g}{G}}
```

Dans cette documentation, on a composé plusieurs méthodes comme exemples de fonctionnement de la classe (pages 31 à 33) ou pour expliquer une possibilité de personnalisation de la classe et d'autres.

Un appel à cette macro va donner le résultat suivant sur une nouvelle page avec la mise en page générale des annexes.

LISTE DES MÉTHODES

Manuel

▶ Créer de nouveaux thèmes	8	▶ Titre de la méthode	32
▶ Créer un nouvel environnement dans les pages Activités	28	▶ Titre de la méthode	32
▶ Titre de la méthode	31	▶ Dernière méthode	33

À noter que comme les méthodes de cette documentation ont été réalisées dans un chapitre avec le thème personnalisé `manuel` il a fallu redéfinir le contenu de `\ListeMethodesThemes` avec la commande :

```
\renewcommand\ListeMethodesThemes{{m}{M}}
```

Voir page 40 où l'on présente la commande `\NewThema` pour les explications des lettres « m » et « M ».

Dans un ouvrage réel, les bandeaux de chaque thème ne seront affichés que s'il y a au moins une méthode dans le thème et, évidemment, si les lettres du thème sont listées dans la macro `\ListeMethodesThemes`.





12. Solutions

La classe automatise la composition des solutions des exercices d'un ouvrage. Lorsqu'on veut qu'un exercice ait une correction, il suffit de le faire suivre immédiatement d'un environnement `corrige`. Ensuite, la seule chose à faire est d'appeler la commande `\AfficheCorriges`.

SYNTAXE :

```
\AfficheCorriges[nombre colonnes]
```

L'argument optionnel indique le nombre de colonnes sur lesquelles composer les solutions aux exercices. Par défaut, c'est la valeur de la macro `\NombreColonnesCorriges` et par défaut, la classe définit cette dernière comme valant 3.

■ Tout se fait automatiquement :

- Chaque chapitre a son bandeau avec son titre et la couleur de son thème.
- À l'intérieur de chaque chapitre, chaque partie à son titre si elle contient au moins une solution à un exercice.
- Si une partie ne contient aucune solution, elle n'aura pas de titre de partie.

REMARQUE : On a utilisé cette commande à la toute fin de cette documentation.

- On voit que le premier chapitre (Introduction) ne contenait aucune solution. C'est un cas qui ne devrait pas arriver en pratique mais cela permet de voir qu'il n'y a rien d'afficher : aucune correction et aucune partie.
- Le deuxième chapitre (Classe pour un ouvrage) contient des solutions d'exercices à la fois dans la partie « Auto-évaluation », « S'entraîner » et « Auto-évaluation » en QCM donc la liste de corrections contiendra ces trois titres de parties.
- Le chapitre suivant (Classe pour quelques feuilles) contient la solution de deux exercices dans la partie Auto-évaluation.
- Enfin, le dernier chapitre (Détails typographiques) ne contient que trois solutions d'exercices dans la partie Auto-évaluation.

ATTENTION : Comme de nombreux éléments de la maquette, une solution d'exercice est insécable. Lorsqu'une solution est particulièrement longue, cela peut devenir disgracieux, voire carrément fautif si la correction prend plus que la hauteur d'une colonne. La maquette incite à des solutions très courtes.

S'il n'est vraiment pas possible de faire autrement, la classe offre la commande `\Coupe` qui permet une coupure de colonne à ce niveau.



■ La commande `\Coupe`

- Cette commande ne provoque pas nécessairement de coupure de colonne mais provoque obligatoirement un fin de paragraphe. La commande `\Coupe` provoque en fait la fermeture de la minipage en cours puis la réouverture d'une autre minipage avec une commande `\par` entre les deux. Cela empêche d'avoir d'autres environnements qui contiendraient cette commande `\Coupe`. En pratique, ce qui risque d'arriver assez souvent est que la commande coupe se trouve à l'intérieur d'une liste, dans le cas le plus embêtant, à l'intérieur d'une liste numérotée. Voici la méthode générale pour gérer ce cas de figure.

■ Le code

```
\begin{corrige}
  \begin{enumerate}
    \item premier point
    \item deuxième point
  \end{enumerate}
  \Coupe
  \begin{enumerate}
    \setcounter{enumi}{2}
    \item troisième point après la coupure
  \end{enumerate}
\end{corrige}
```

ATTENTION : Cette commande est un peu délicate à utiliser et sert à corriger les coupures de colonnes d'un ouvrage. Comme tout ce qui concerne les coupures de page ou de colonne, c'est à utiliser en toute fin de composition, une fois qu'on est absolument certain que le texte ne changera plus... sinon, vous risquez de travailler pour rien ! C'est une commande pour le metteur en page, pas pour l'auteur (si les deux sont différents).

■ Organisation pratique

Chaque fois que l'environnement `corrige` est utilisé, la classe crée un fichier à part avec la correction. Ces fichiers seront appelés lors de la création de la page Solution à l'appel de la commande `\AfficheCorriges`.

Si l'environnement est souvent utilisé, cela va créer une multitude de petits fichiers à la racine du répertoire.

Il est toutefois possible de créer un répertoire « corrections » et de demander à la classe d'y ranger les fichiers de correction avec la commande suivante à mettre avant le `\begin{document}`.

SYNTAXE : `\renewcommand\PrefixeCorrection{corrections/}`





13. Propriétés

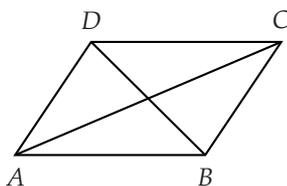
Une annexe à l'allure tout à fait particulière est celle dédiée aux propriétés. En réalité, il s'agit d'une annexe générale classique, c'est-à-dire introduite avec la commande `\annexe` mais la classe offre plusieurs commandes et environnements particuliers pour gérer cette présentation.

En premier lieu, il y a une syntaxe très particulière de tableau. Les tableaux seront présentés à la section D page 58 et on pourra s'y référer pour plus de détails sur la syntaxe générale. Celui qui va servir à composer les propriétés est l'environnement `tableau` avec l'argument optionnel `pr`. Cet environnement demande un argument obligatoire indiquant la largeur totale du tableau. Il compose des tableaux sur trois colonnes avec la colonne du milieu légèrement plus large que les deux autres et avec un fond coloré. Théoriquement, si on respecte la maquette, la première colonne est là pour accueillir la figure illustrant la propriété, la deuxième colonne pour le texte de la propriété et la troisième pour indiquer le lien entre la figure et la propriété.

Le code

```
\begin{tableau}[pr]{\linewidth}
  \hline
  \begin{pspicture}(0,0.25)(3.5,2.5)
    \pnode(0,0.5){A}
    \pnode(2.5,0.5){B}
    \pnode(3.5,2){C}
    \pnode(1,2){D}
    \pspolygon(A)(B)(C)(D)
    \psline(A)(C)
    \psline(B)(D)
    \uput[d](A){$A$}
    \uput[d](B){$B$}
    \uput[u](C){$C$}
    \uput[u](D){$D$}
  \end{pspicture}
  &
  \propriete{} Si un quadrilatère est un parallélogramme alors ses
  diagonales se coupent en leur milieu. (C'est aussi vrai pour les
  losanges, rectangles et carrés qui sont des parallélogrammes
  particuliers.)
  &
  Ici $ABCD$ est un parallélogramme donc ses diagonales $[AC]$ et
  $[BD]$ se coupent en leur milieu.
  \\ \hline
\end{tableau}
```

donne le résultat :



PROPRIÉTÉ 1 Si un quadrilatère est un parallélogramme alors ses diagonales se coupent en leur milieu. (C'est aussi vrai pour les losanges, rectangles et carrés qui sont des parallélogrammes particuliers.)

Ici $ABCD$ est un parallélogramme donc ses diagonales $[AC]$ et $[BD]$ se coupent en leur milieu.

Cet exemple montre l'utilisation de la commande `\propriete` qui compose le mot « PROPRIÉTÉ » suivi de son numéro dans un cadre en couleur. À chaque appel de cette macro, le numéro augmentera.

REMARQUE : Il est possible d'utiliser le mécanisme des `\label` pour faire référence aux propriétés de ce type de tableau.

La classe définit également la commande `\ListeProprietes` qui permet de retrouver le même cartouche que celui de la commande `\propriete` mais avec un arguments donnant la liste des numéros.

Le code

```
\ListeProprietes{P1 à P6}
```

donne le résultat :

PROPRIÉTÉ P1 à P6 .

Cela permet de donner des explications générales sur des regroupements de propriétés. On peut également utiliser la commande `\section` pour séparer les propriétés en plusieurs thèmes généraux.

14. Lexique

La classe `sesamaneul` permet de composer un lexique ou glossaire de façon totalement automatique, c'est-à-dire sans recours à aucun outil externe, et de façon très souple.

Pour inclure un objet dans le lexique, on utilise la commande `\MotDefinition` dont la syntaxe est très riche :

SYNTAXE : `\MotDefinition[lexique]{mot}[tri]{définition}`

La commande a donc deux arguments obligatoires. Le premier est le mot ou l'expression à mettre dans le lexique et le second est la définition de ce mot ou de cette expression. On verra la signification des arguments optionnels après.

Le deuxième argument peut être vide ou non. S'il n'est pas vide, on écrit le mot suivi de sa définition dans le glossaire. S'il est vide, on écrit le mot dans le glossaire suivi de la page où la commande `\MotDefinition` a été utilisée et le mot est composé en gras dans le texte. Voici un premier exemple.

Le code

```
\MotDefinition{glossaire}{Liste de mots utilisés dans le texte avec leurs définitions}
Dans cette section, on parle de \MotDefinition{lexique}{} et de glossaire.
```

donne le résultat :

Dans cette section, on parle de **lexique** et de glossaire.



On peut voir que les entrées avec une définition peuvent être saisies à n'importe quel endroit du document, voire regroupées dans un fichier externe appelé par un `\input`.

REMARQUE : Dans certains cas, le terme apparaissant dans le lexique ne peut pas être classé alphabétiquement de façon satisfaisante. Par exemple, supposons qu'on veuille indexer le symbole \int , le mot **intégrale** ainsi que le mot **intégration** mais qu'on veuille que, dans le lexique, le symbole mathématique soit classé immédiatement après le mot « intégrale ». On se sert alors du deuxième argument optionnel qui permet de spécifier le classement alphabétique. Par défaut, le classement s'effectue en se servant du premier argument obligatoire sauf si, justement, on utilise le deuxième argument optionnel.

Le code

```
\MotDefinition{\int}[integralea]{}
\MotDefinition{intégrale}{}
\MotDefinition{intégration}{}

```

va produire trois entrées dans le lexique : les deux mots « intégrale » et « intégration » et le symbole « \int » classé immédiatement après le mot « intégrale ».

Le premier argument optionnel n'a de sens que lorsqu'on ne donne pas de définition, c'est-à-dire lorsque le deuxième argument obligatoire est vide. Il sert à avoir un mot différent entre ce qui est placé dans le lexique et ce qui est composé dans le texte. Normalement, ce sont les mêmes **mots** qui servent pour les deux usages mais, de temps en temps, on voudrait faire référence à un mot qui n'est pas exactement celui employé dans le texte. Un exemple fréquent est l'utilisation d'un mot au pluriel dans le texte alors que le lexique présente normalement le mot au singulier. Par exemple, on a utilisé le code :

```
\MotDefinition[mot]{mots}{}

```

un peu au-dessus dans ce paragraphe. Cela a fait que c'est bien « mots » (au pluriel) qui a été mis en gras dans ce paragraphe mais que c'est « mot » (au singulier) qu'on peut voir dans l'entrée d'index ci-dessous.

LEXIQUE

G

Glossaire

Liste de mots utilisés dans le texte avec leurs définitions

I

Intégrale Page 45

\int Page 45
Intégration Page 45

L

Lexique Page 44

M

Mot Page 45



L'affichage du lexique se fait grâce à la commande `\AfficheLexique`. Sa syntaxe est semblable à celle de la commande `\AfficheCorriges`.

SYNTAXE :

```
\AfficheLexique[nombre colonnes]
```

L'argument optionnel indique le nombre de colonnes sur lesquelles composer le lexique. Par défaut, c'est la valeur de la macro `\NombreColonnesLexique` et par défaut, la classe définit cette dernière comme valant 2.

REMARQUE : La classe procède à plusieurs traitements automatiques.

- En premier lieu le classement alphabétique (en traitant les accents comme la lettre non accentuée) et la mise en minuscule de l'entrée avec la première lettre en capitale. Si ces automatismes doivent être inhibés, il faudra utiliser les arguments optionnels de la commande `\MotDefinition`.
- En deuxième lieu le placement d'un bandeau pour chaque groupe de première lettre existant.

C'est la classe elle-même qui s'occupe d'un classement alphabétique. Pour cela, si l'argument optionnel de tri n'est pas spécifié, c'est le mot en lui-même qui sera utilisé pour le tri. La classe procède à un traitement préalable pour fabriquer sa clé de tri à partir du mot. Ce traitement consiste à traduire en minuscule en changeant toutes les lettres diacritiques en la ou les lettres simples correspondantes (« ç » en « c », « ê » en « e », « œ » en « oe », etc.). Cette façon de faire ne donne pas strictement l'ordre alphabétique français puisque les diacritiques dépendant d'une seule lettre sont classées au hasard. Par exemple, les mots « côté » et « coté » seront classés de façon aléatoire. Pour un lexique d'ouvrage mathématique, le cas ne devrait pas arriver souvent et, au pire, on peut y remédier avec l'argument optionnel de tri mais c'est néanmoins une petite amélioration à laquelle il faudra songer pour une prochaine version de la classe.



Classe pour quelques feuilles

Connaissances du collège nécessaires à ce chapitre

Dans ce chapitre, nous présentons l'utilisation de la classe permettant de construire quelques éléments d'un ouvrage sans devoir en suivre l'organisation générale.

Typiquement, cette partie de la classe a été pensée pour qu'une personne puisse aller sur le site de Sesamath, prendre le code source d'un ou plusieurs exercices (par exemple) et compiler un document ne contenant que ces quelques exercices.

En prenant quelques éléments épars de la classe on a construit l'extension `sesamanuel.sty`. Celle-ci permet de compiler des exemples simples qui utilisent quelques éléments de syntaxe de la classe `sesamanuel` mais sous une autre classe comme `beamer`.

Une première section étudiera l'utilisation de la classe et une seconde l'utilisation de l'extension.



Auto-évaluation

Des ressources numériques pour préparer le chapitre sur manuel.sesamath.net



1 Où trouver les codes sources des manuels ?

2 Je voudrais faire une présentation de mes exercices avec beamer. Comment procéder ?

▶▶▶ Voir solutions p. 74



1. Utilisation de la classe

Pour utiliser la classe dans le but de produire une page (ou peu de pages), il existe un pseudo thème dit « libre ». En fait, on ne l'appelle même pas directement. Pour indiquer qu'on va utiliser la classe de façon « libre », on indiquera la macro `\libre` en tout début de document.

Cette commande va considérablement simplifier la mise en page des en-têtes (devenues inexistantes) et des pieds de page (réduits au seul numéro de page).

La conséquence immédiate de cela est qu'il devient alors sans objet de vouloir utiliser les commandes de partie générale d'un ouvrage (thème, chapitre, partie dédiée aux activités d'approche, au cours, aux exercices, aux QCM d'évaluation et aux TP). En revanche, on peut quasiment utiliser tous les éléments internes de la classe.

Il existe quelques limitations dues à l'absence de cloisonnement de parties générales. Ainsi, la commande `\partie` peut exister à plusieurs emplacements avec un traitement différents dans un ouvrage classique. Ce comportement ne pourra plus exister puisque les commandes modifiant le comportement de `\partie` ne seront plus utilisées.

Voici ce qu'il est possible de faire avec une utilisation « libre » de la classe :

- écrire du texte de façon libre ;
- composer un environnement `debat` ou `activite` ;
- utiliser la commande `\section` (non numérotées) ;
- utiliser les environnements de type `definition`, `notation`, `methode` (avec la liste des méthodes si on le souhaite en appelant la commande `\AfficheListeMethodes`) ;
- utiliser la commande `\recreation` (mais par la commande `\Recreation` qui modifie les en-têtes) ;
- les QCM d'auto-évaluation ;
- les TP ;
- les exercices (et le mécanisme des corrections) ;
- le lexique.

Tout ce qui a été dit dans le chapitre précédent s'applique. On notera juste l'introduction de quelques paramètres propres à la partie « libre ».

Couleurs dans la partie libre

`LibrePartieColor` A1 Couleur des titres de partie `\algo`

Les autres couleurs sont gérées par la commande `\ChangeLibre` qui a exactement la même syntaxe et la même signification que la commande `\ChangeAnnexe`.

Il y a une certaine homogénéité dans le choix des couleurs à l'intérieur d'une partie libre. Il est possible, comme d'habitude, de modifier les couleurs de façon individuelle mais, pour assurer cette homogénéité la classe fournit la commande `\ChangeLibre` qui demande quatre arguments qui seront les couleurs utilisées dans la partie libre. Le premier argument correspond à la couleur du bandeau de marge mais dans la partie libre, il n'y a pas de tel bandeau. On garde ce premier argument pour une éventuelle modification future mais pour l'instant, il ne sert à rien. Le deuxième argument est une couleur partagée par plusieurs éléments :

- `LibreSectionRuleColor`
- `LibreItemColor`
- `LibreExerciceCorrigeNumFrameColor`
- `LibreExerciceNumFrameColor`

Le troisième argument est également une couleur partagée par plusieurs éléments :

- `LibreSectionTitleColor`





- LibreExerciceTitleColor

Enfin, le quatrième et dernier argument est la couleur des numéros d'exercices. Par défaut la classe appelle cette macro de cette façon, c'est-à-dire exactement comme pour les annexes :

```
\ChangeLibre{G3}{A1}{G1}{Blanc}
```

On a également quelques définitions de longueurs :

Longueurs dans la partie libre	
<code>\LibreSectionRuleWidth</code>	<code>\SquareWidth</code>
<code>\LibreSectionRuleHSpace</code>	1mm
<code>\BeforeLibreSectionVSpace</code>	5mm
<code>\AfterLibreSectionVSpace</code>	2mm
<code>\BeforeLibrePartieVSpace</code>	1.5mm
<code>\AfterLibrePartieVSpace</code>	1mm

Enfin, quelques définitions de fontes :

Fontes dans la partie libre	
<code>\FootLibrePageFont</code>	<code>\fontsize{11}{14.4}\sffamily\bfseries</code>
<code>\LibreSectionTitleFont</code>	<code>\fontsize{14.4}{14.4}\sffamily\bfseries</code>
<code>\LibrePartieFont</code>	<code>\fontsize{10.5}{14.4}\bfseries</code>

2. Extension `sesmanuel`

L'extension `sesmanuel` est destinée à être utilisée lorsqu'il est obligatoire d'utiliser une autre classe que `sesmanuel`. Un exemple typique est celui où on compose un diaporama avec la classe `beamer`.

L'extension définit un certain nombre d'éléments et de syntaxe qui sont définis dans la classe mais un grand nombre de mécanismes sont évidemment abandonnés. Tout ce qu'on trouve ici est également utilisable avec une utilisation « libre » de la classe (voir ci-dessus).

Avec l'extension, on a accès à :

- les tableaux ;
- les algorithmes ;
- les unités ;
- les macros `\TopStrut` et `\BotStrut` ;
- les listes.

Tous ces éléments sont présentés plus en détails dans le chapitre suivant.

Pour les listes, comme la classe prévoyait un grand nombre de possibilités, on les retrouve dans l'extension grâce à la macro `\setlist`. Celle-ci demande un argument qui indique le type de partie donc l'aspect des différentes listes (voir la première section du chapitre suivant). On trouve :

- `\setlist{prerequis}` ;
- `\setlist{AE}` pour les listes d'auto-évaluation ;



- `\setlist{exapp}` pour les listes des exercices d'application ;
- `\setlist{def}` pour les listes des environnements de type définition ;
- `\setlist{activite}` ;
- `\setlist{rem}` pour les listes des environnements de type remarque ;
- `\setlist{exemple}` ;
- `\setlist{proof}` ;
- `\setlist{exercice}` pour les listes des exercices d'entraînement et d'approfondissement ;
- `\setlist{acquis}` pour les listes dans « Je vérifie mes acquis » ;
- `\setlist{commentaire}` ;
- `\setlist{recreation}` ;
- `\setlist{annexe}` ;
- `\setlist{lexique}`.

L'utilisation des listes dans une classe générale requiert quelques précautions.

En premier lieu il est obligatoire de spécifier un des types de liste précédents avec la macro `\setlist`.

Les environnements `colenumerate` et `colitemize` se servent de la macro `\item` pour analyser le contenu d'une liste afin de le découper en différents blocs. Si vous emboîtez des listes dans ce type de liste, il va falloir protéger la liste intérieure en la plaçant intégralement dans un groupe.

Enfin, il est fortement préférable de travailler avec des documents sans retrait d'alinéa sinon le rendu risque d'être assez bizarre.

Voici un code qui tient compte de tous ces éléments :

```
\setlength{\parindent}{0pt}
\setlist{recreation}
\begin{colenumerate}
\item Le premier point est simple.
\item Le deuxième point contient une liste (protégée) :
{
  \begin{enumerate}
  \item avec un premier sous-point
  \item et un deuxième.
  \end{enumerate}
}
\item Le troisième point est simple
\end{colenumerate}
```

Détails typographiques

Constructions offertes par la classe

Dans ce chapitre, nous allons présenter les éléments qui existent pour toutes les parties du manuel ou, du moins, pour plusieurs parties du manuel. On découpera la présentation en :

► Listes

Il s'agit des listes traditionnelles `itemize` et `enumerate` avec leurs différentes déclinaisons selon l'endroit où elles sont appelées.

► Cadres et tableaux

La classe offre un certain nombre de constructions automatiques de tableaux ainsi que des cadres variés.

► Divers

On placera dans cette dernière section tout ce qui ne relève pas des deux sections précédentes. Cela fera donc un peu fourre-tout.



Auto-évaluation

Des ressources numériques pour préparer le chapitre sur manuel.sesamath.net



1 Essayez de placer des environnements `itemize` ou `enumerate` dans plusieurs endroits du document et regardez le rapport de compilation.

2 Comment faire pour commencer la partie cours en première page d'un chapitre ?

3 Comment procéder pour réaliser un manuel de physique à la place d'un manuel de mathématique ?

▶▶▶ Voir solutions p. 74



1. Listes

La classe redéfinit les listes standards `itemize` et `enumerate`. Elle ne redéfinit pas la liste `description` puisque la maquette initiale ne s'en servait jamais. Elle crée également d'autres listes, à savoir `colitemize`, `colenumerate` et `bigenumerate`.

Les environnements `colitemize` et `colenumerate` permettent de présenter les résultats en colonnes. Le principe est de spécifier le nombre de colonnes voulues en paramètre. Les entrées des listes se feront alors en remplissant les colonnes d'abord et en les équilibrant au maximum. Voici un exemple. Le source

```
\begin{colenumerate}{3}
\item item
\item item
\item item un peu plus long pour faire plus d'une ligne
\item item un peu plus long pour faire plus d'une ligne
\item item
\end{colenumerate}
```

donne le résultat :

- | | | |
|---------|--|---------|
| 1) item | 3) item un peu plus long pour faire plus d'une ligne | 5) item |
| 2) item | 4) item un peu plus long pour faire plus d'une ligne | |

On aura un résultat similaire pour les listes `itemize`.

La liste `bigenumerate` résulte d'une discussion lors de l'élaboration de la maquette. L'exemple précédent montre que les numéros d'une liste `enumerate` ou `colenumerate` sont très proches du texte de l'item. Si le numéro vient à s'écrire sur deux chiffres, soit il télescope le texte, soit il chasse dans la marge, soit le texte de l'entrée n'est plus aligné avec celui des autres entrées. C'est cette dernière option qui a été choisie mais avec la possibilité d'avoir une liste permettant quand même l'alignement en augmentant la distance entre numéro et texte. Voici un exemple (idiot puisque `bigenumerate` ne sert qu'à partir de 10 entrées). Le source :

```
\begin{enumerate}
\item item
\item item
\end{enumerate}
\begin{bigenumerate}
\item item éloigné
\item item éloigné
\end{bigenumerate}
```

donne le résultat :

- | | |
|-----------------|--|
| 1) item | |
| 2) item | |
| 1) item éloigné | |
| 2) item éloigné | |





La classe prévoit deux niveaux de liste. On peut en faire plus mais l'aspect restera celui du deuxième niveau d'emboîtement. Pour les liste `itemize`, cela ne se traduira que pas un retrait différent. Pour les listes `enumerate`, on aura une présentation différentes des numéros. Voici un exemple :

```
\begin{enumerate}
\item item
\item item
  \begin{enumerate}
  \item item emboîté
  \item item emboîté
  \end{enumerate}
\item item
\end{enumerate}
```

qui donne :

- 1) item
- 2) item
 - a) item emboîté
 - b) item emboîté
- 3) item

En réalité la façon dont sont composées les listes dépend de l'emplacement où elles sont appelées. Les exemples ci-dessus ont été construits avec les listes utilisées dans l'auto-évaluation (bas de la première page d'un chapitre). On va montrer toutes les possibilités avec le source :

```
\begin{itemize}
\item test itemize
\end{itemize}
\begin{enumerate}
\item niveau 1
  \begin{enumerate}
  \item niveau 2
  \end{enumerate}
\end{enumerate}
```

Listes dans le cadre bleu des prérequis

- ▶ test itemize
- 1) niveau 1
 - a) niveau 2

Listes dans les activités et débats

- test itemize
- 1) niveau 1
 - a) niveau 2

Listes dans les environnements de type remarque

- test itemize
- 1) niveau 1
 - a) niveau 2

Listes dans les QCM d'auto-évaluation

- test itemize
- 1) niveau 1
 - a) niveau 2

Listes dans les environnements de type définition

- test itemize
- 1) niveau 1
 - a) niveau 2

Listes dans les environnements de type preuve

- test itemize
- 1) niveau 1
 - a) niveau 2



Listes dans les exemples

- test itemize
- 1) niveau 1
- a) niveau 2

Listes dans les commentaires

- test itemize
- 1) niveau 1
- a) niveau 2

Listes dans les questions de je teste mes connaissances

- ▶ test itemize
- 1) niveau 1
- a) niveau 2

Listes dans les correction des QCM d'auto-évaluation

- test itemize
- 1) niveau 1 a) niveau 2

Listes dans les corrections des récréations

- test itemize
- 1) niveau 1 a) niveau 2

Listes dans le lexique

- test itemize
- 1) niveau 1
- a) niveau 2

L'utilisation de ces listes en d'autres emplacements du document provoquera une erreur de compilation. Ainsi, on ne pourra pas utiliser ces listes :

- dans les pages activités (en dehors des activités et débats proprement dits) ;
- dans les pages cours (en dehors de certains environnements) ;
- dans les QCM de je teste mes connaissances ;
- dans les corrections d'activités ;
- dans les corrections des je teste mes connaissances.

Listes dans les méthodes

- test itemize
- 1) niveau 1
- a) niveau 2

Listes dans les exercices

- test itemize
- 1) niveau 1
- a) niveau 2

Listes dans les récréations et énigmes

- test itemize
- 1) niveau 1
- a) niveau 2

Listes dans les corrections d'exercices

- test itemize
- 1) niveau 1 a) niveau 2

Listes dans les annexes générales

- test itemize
- 1) niveau 1
- a) niveau 2

2. Cadres et tableaux

A. Cadres

Il y a plusieurs cadres qui sont définies par la classe mais *via* des macros privées. L'utilisateur ne peut donc normalement pas s'en servir.

Si on excepte les cadres vus au chapitre précédent qui devaient être placés à des endroits particuliers du manuel, on trouve une commande particulière permettant de composer un cadre avec un biseau inférieur droit.

Ce type de cadre est réalisé grâce à l'environnement `cadre`. Cet environnement accepte deux arguments optionnels indiquant la couleur du cadre et la couleur de fond. Par défaut, ce sera respectivement la couleur `CadreLineColor` et la couleur `CadreBgColor`. La largeur des lignes est indiquée avec la macro `\CadreLineWidth` et la séparation





```

1. Pour x dans 1..n //entrée de boucle
2.   Pour y dans 1..n
3.     M[x,y]=x+y
4.   Fin Pour
5. Fin Pour
    
```

La classe propose un autre environnement pour réaliser des algorithmes qui ne suit pas du tout le même principe que celui ci-dessus. L'environnement `algorithme` permet de présenter des algorithmes dans un cadre en utilisant des commandes dédiées qui se chargeront automatiquement du retrait des lignes. Les commandes utilisables sont :

Commande	Signification
<code>\BlocVariables</code>	Affiche « Liste des variables utilisées »
<code>\BlocEntrees</code>	Affiche « Entrées »
<code>\BlocTraitements</code>	Affiche « Traitements »
<code>\BlocAffichage</code>	Affiche « Affichage »
<code>\BlocTraitementsEtAffichage</code>	Affiche « Traitements et affichage »
<code>\DeclareVar</code>	Demande 3 paramètres qui sont la variable, son type et un commentaire
<code>\TantQue</code>	Demande 2 paramètres qui sont la condition de boucle et le corps de boucle
<code>\Pour</code>	Demande 4 paramètres qui sont la variable de boucle, la valeur initiale, la valeur finale et le corps de boucle
<code>\PourAvecPas</code>	Demande 5 paramètres qui sont la variable de boucle, la valeur initiale, la valeur finale, le pas et le corps de boucle
<code>\SiAlors</code>	Demande 2 paramètres qui sont la condition et la partie vraie
<code>\SiAlorsSinon</code>	Demande 3 paramètres qui sont la condition, la partie vraie et la partie fausse
<code>\AfficherVar</code>	Demande 1 paramètre qui est la variable
<code>\Afficher</code>	Demande 1 paramètre qui est ce qu'il faut afficher
<code>\Calculer</code>	Demande 1 paramètre qui est le calcul
<code>\Demander</code>	Demande 1 paramètre qui est la demande
<code>\Saisir</code>	Demande 1 paramètre qui est ce qui doit être saisi
<code>\Stocker</code>	Demande 1 paramètre qui est la variable dans laquelle on stocke
<code>\DonnerValeur</code>	Demande 2 paramètres qui sont le nom d'une variable et la valeur à stocker
<code>\TitreAlgo</code>	Demande 1 paramètre qui est le titre de l'algorithme
<code>\FinAlgo</code>	Affiche « Fin de l'algorithme »

Chaque instruction gère elle-même l'indentation, les fontes et les couleurs utilisées. Le tableau 2.2 pages 10-16 indique les noms de couleurs utilisées pour les algorithmes (ce sont les noms qui commencent par `CouleurAlgo`). En fait, toutes ces couleurs sont noires mais la classe permet une coloration syntaxique automatisée personnalisée. Voici un exemple d'algorithme pour mieux comprendre le fonctionnement de l'indentation automatique. Celui-ci est sous le contrôle de la macro `\AlgoIndent` qui vaut `1em` par défaut. Le code

```

\begin{algorithme}
  \TitreAlgo{Euclide}
  \BlocVariables
  \DeclareVar{a}{entier}{1\ier{ } nombre}
  \DeclareVar{b}{entier}{2\ieme{ } nombre}
  \DeclareVar{r}{entier}{reste}
  \BlocEntrees
    
```



```
\Demander{a}
\Demander{b}
\BlocTraitements
\Calculer{r = reste de a/b}
\TantQue{r $\neq$ 0}{%
  \DonnerValeur{a}{b}
  \DonnerValeur{b}{r}
  \Calculer{r = reste de a/b}
}
\Afficher{"PGCD = "}
\AfficherVar{b}
\FinAlgo
\end{algorithm}
```

donne le résultat :

1. *Algorithme* : Euclide
2. *Liste des variables utilisées*
3. a : *entier* 1^{er} nombre
4. b : *entier* 2^e nombre
5. r : *entier* reste
6. *Entrées*
7. Demander a
8. Demander b
9. *Traitements*
10. Calculer r = reste de a/b
11. **Tant que** (r \neq 0) **faire**
12. Donner à a la valeur de b
13. Donner à b la valeur de r
14. Calculer r = reste de a/b
15. **Fin Tant que**
16. Afficher "PGCD = "
17. Afficher la valeur de b
18. *Fin de l'algorithme*

C. Sudokus

La classe a une commande très particulière qui permet de composer des sudokus. Il s'agit de la commande `\sudoku`. Cette commande accepte un argument optionnel qui indique la largeur d'une case (`SudokuWidth` par défaut, c'est-à-dire 5 mm) et un argument obligatoire qui indique le contenu de la grille de sudoku.

Une grille de sudoku est un tableau 9×9 et on va donc donner 9×9 indications de remplissage. Un point représente une case vide blanche, une étoile représente une case vide colorée (avec la couleur `FondSudokuColor`) et toute autre matériel indique ce qu'il faut placer dans la case (en général des chiffres).

En réalité, la commande est assez permissive. S'il y a moins de 81 renseignements, la fin de la grille sera remplie avec des cases vides. S'il y a plus de 81 renseignements, les indications après la 81^e seront ignorées. Les espaces et les sauts de ligne ne comptent pas ce qui permet de noter les grilles de façon lisible pour l'humain.



Le code

```
\begin{center}
\sudoku{}
\hspace{2cm}%
\sudoku{%
1.234..5.
.....
5..67.8..
6..5.2..8
23..*..61
9..1.7..4
..4.16..3
.....
.2..897.5
}
\end{center}
```

donne le résultat :

	a	b	c	d	e	f	g	h	i
A									
B									
C									
D									
E									
F									
G									
H									
I									

	a	b	c	d	e	f	g	h	i
A	1		2	3	4			5	
B									
C	5			6	7		8		
D	6			5		2			8
E	2	3						6	1
F	9			1		7			4
G			4		1	6			3
H									
I		2			8	9	7		5

D. Tableaux

La classe permet d'utiliser toutes les commandes et environnements standards liés aux tableaux. Elle charge même les extensions `longtable` donc `array`, `tabularx`, `multirow` et l'extension `xcolor` avec son option `table`.

Cependant, pour assurer une cohérence dans le traitement des tableaux, la classe propose une série d'environnements permettant de présenter des tableaux sous des formes particulières.

Les environnements s'appellent `<type>tableau`. À l'intérieur de cet environnement, la couleur des filets sera grise. Le type `X` de l'environnement `tabularx` est redéfini pour donner une entrée centrée à la fois verticalement et horizontalement. Le `<type>` est une ou deux lettres indiquant le type du tableau, on a les possibilités suivantes :





<type>	Signification
t	C'est l'environnement le plus simple. Il demande deux arguments qui sont la largeur du tableau et le nombre de colonnes. Toutes les colonnes seront du type X
c	Même chose que ci-dessus mais la première colonne sera en couleur.
l	Même chose que ci-dessus mais c'est la première ligne qui sera en couleur.
cl	Toujours la même chose. C'est la première ligne et la première colonne qui seront en couleur.
T, C, L et CL	Les types en capitale correspondent au type en minuscule. Il demande un argument supplémentaire qui est le type de la première colonne.
pr	Il s'agit d'un type de tableau très particulier qui ne sert, normalement, que pour la liste des propriétés (voir page 13)

Voici un exemple avec le type C. Le source

```
\begin{center}
\renewcommand*\tabularxcolumn[1]{>{\centering\arraybackslash}m{#1}}
\begin{Ctableau}{0.7\linewidth}{7}{c}
\hline
Hauteur en dm & 0,5 & 1 & 1,5 & 2 & 2,5 & 3 \\ \hline
Volumes en litres & & & & & & \\ \hline
\end{Ctableau}
\end{center}
```

va donner le résultat :

Hauteur en dm	0,5	1	1,5	2	2,5	3
Volumes en litres						

Dans l'exemple, on a redéfini la macro `\tabularxcolumn` afin d'obtenir des entrées centrées.

Lorsqu'on compose un tableau avec un des types `c`, `l`, `cl` et les types en capitale correspondants, on utilise la couleur `FondTableaux`. La couleur des filets est `FiletTableauColor`

3. Divers

Dans cette section, on va présenter des détails typographiques définis par la classe. On ne parlera pas de tous les détails typographiques définis par les extensions appelées par la classe en renvoyant à la documentation de ces extensions (liste page 5).

A. Empagement

Dans un ouvrage *Sésamath*, l'empagement peut être modifié dans certaines circonstances (à l'intérieur des activités ou débats, dans les travaux pratiques).

Normalement, l'utilisateur n'a pas à régler lui-même ces problèmes d'empagement. Cela dit, la classe laisse quand même la possibilité de le faire en fournissant l'environnement `changemargin`. Celle-ci est utilisé en interne pour certaines présentation où l'empagement est réduit. Si l'augmentation de marge à gauche dépend des circonstances, l'augmentation de la marge droite est toujours égale à `\ExtraMarginRight`.

Lorsque l'empagement est réduit, il peut alors arriver qu'on veuille remettre l'empagement jusqu'à la marge droite habituelle. Pour cela, on utilise l'environnement `debordedroite`. Par exemple, le code :



```
\begin{changemargin}{1.5cm}{\ExtraMarginRight}
  Exemple de texte avec une modification d'empagement...
  \begin{debordedroite}
    Exemple de texte avec une marge droite habituelle...
  \end{debordedroite}
  Exemple de texte avec une modification d'empagement...
\end{changemargin}
```

donnera le résultat suivant :

Exemple de texte avec une modification d'empagement. Exemple de texte avec une modification d'empagement.

Exemple de texte avec une marge droite habituelle. Exemple de texte avec une marge droite habituelle. Exemple de texte avec une marge droite habituelle. Exemple de texte avec une marge droite habituelle.

Exemple de texte avec une modification d'empagement. Exemple de texte avec une modification d'empagement. Exemple de texte avec une modification d'empagement. Exemple de texte avec une modification d'empagement.

Un autre commande permettant de jouer ponctuellement avec l'empagement est celle qui permet d'habiller une image avec du texte. Dans les ouvrages *Sésamath*, les images sont toujours utilisées dans les passages à marge augmentée. Lorsqu'une image est insérée, et si sa largeur le permet, elle débordera pour aller jusqu'à la marge de droite habituelle. Voici un exemple où on va inclure les deux cas de figure possibles (images plus large ou moins large que `\ExtraMarginRight`) :

```
\begin{changemargin}{1.5cm}{\ExtraMarginRight}
  \setkeys{Gin}{keepaspectratio=false, height=2cm}
  \habillage{\fbox{\includegraphics[width=1cm]{tiger.eps}}} Dans ce premier...

  \habillage{\fbox{\includegraphics[width=4cm]{tiger.eps}}} Dans ce second...
\end{changemargin}
```

qui donne le résultat :

Dans ce premier exemple, l'image a une largeur inférieure à la marge supplémentaire `\ExtraMarginRight`. Dans ces conditions, l'image débord de sa largeur naturelle sans renforcement dans le texte. Dans ce premier exemple, l'image a une largeur inférieure à la marge supplémentaire `\ExtraMarginRight`. Dans ces conditions, l'image débord de sa largeur naturelle sans renforcement dans le texte.

Dans ce second exemple, la largeur de l'image est maintenant supérieure à `\ExtraMarginRight`. Dans ces conditions, la droite de l'image atteint exactement la marge droite habituelle et provoque un renforcement du texte en vis à vis. Dans ce second exemple, la largeur de l'image est maintenant supérieure à `\ExtraMarginRight`. Dans ces conditions, la droite de l'image atteint exactement la marge droite habituelle et provoque un renforcement du texte en vis à vis.





B. Correction de hauteur

Dans certaines circonstances il peut être utile de poser des éléments invisibles ayant une certaine hauteur. Cela permet par exemple de respecter un interlignage ou bien d'éviter que certains éléments se télescopent (en particulier avec des formules mathématiques complexes). \LaTeX propose la commande `\strut` qui permet de poser un élément ayant exactement la hauteur et la profondeur de l'interlignage classique. La classe étend un peu cette commande en proposant trois commandes : `\Strut`, `\TopStrut` et `\BotStrut`.

La commande `\Strut` (avec une majuscule) fonctionne exactement comme `\strut` mais elle admet un argument optionnel qui est un facteur d'agrandissement. Sa valeur par défaut est 1 ce qui donne donc le comportement exact de `\strut`. Par exemple, le code :

```
\begin{minipage}{0.5\linewidth}
  Il y a une différence fondamentale entre  $\int_0^{+\infty} f(x)dx$ 
  et son équivalent dans le domaine discret qui est la somme infinie
   $\sum_{n=0}^{+\infty} f(n)$ .

  Il y a une différence fondamentale entre  $\int_0^{+\infty} f(x)dx$ 
  et son équivalent dans le domaine discret qui est la somme infinie
   $\Strut[2]\int_0^{+\infty} f(x)dx$ 
  et son équivalent dans le domaine discret qui est la somme infinie
   $\Strut[2]\sum_{n=0}^{+\infty} f(n)$ .
\end{minipage}
```

donne le résultat :

Il y a une différence fondamentale entre $\int_0^{+\infty} f(x)dx$ et son équivalent dans le domaine discret qui est la somme infinie $\sum_{n=0}^{+\infty} f(n)$.

Il y a une différence fondamentale entre $\int_0^{+\infty} f(x)dx$ et son équivalent dans le domaine discret qui est la somme infinie $\sum_{n=0}^{+\infty} f(n)$.

Dans cet exemple, l'utilisation de `\Strut` n'était peut-être pas la meilleure chose à faire si le bas de la formule somme ne gênait pas. Pour permettre une plus grande souplesse dans ce type d'espacement, la classe offre donc les deux commandes supplémentaires `\TopStrut` et `\BotStrut` qui fonctionnent strictement comme `\Strut`, qui donne exactement le même espacement vertical sauf que `\TopStrut` a une profondeur nulle et `\BotStrut` une hauteur nulle. Toujours pour rester dans l'exemple ci-dessus, il est possible que `\TopStrut` ait été un meilleur choix.

C. Unités

La classe offre un système souple pour écrire les nombres avec unités. L'extension `siunitx` est bien plus puissante que ce que propose la classe mais les fonctionnalités desservies par `siunitx` et non proposées par la classe sont de peu d'intérêt pour un ouvrage scientifique du secondaire et, en contre-partie, la syntaxe proposée par la classe est bien plus souple.

L'idée générale est de composer un nombre avec unité en utilisant une macro `\uxxx` ou `\Uxxx` où `xxx` dépend de l'unité voulue. La différence entre la version minuscule et la version majuscule est que celle en minuscule compose un nombre décimal (des chiffres avec un éventuel séparateur décimal) tandis que la version majuscule compose l'argument en mode mathématique. Cela permet de pouvoir composer à la fois $\sqrt{2}$ cm et 1,414 2 cm.

Ces deux derniers exemples ont été produits respectivement par `\Ucm{\sqrt{2}}` et `\ucm{1.4142}`. On peut voir



que même si le source du deuxième exemple comportait un point décimal, la sortie a affiché une virgule. On peut également s'apercevoir que les chiffres ont un séparateur des milliers (ici au niveau de la partie décimale).

Pour la version minuscule, l'argument est en fait composé en tant qu'argument d'une commande `\numprint` de l'extension `numprint`. Toutes les possibilités de modification de paramètre de `numprint` sont donc accessibles. Par exemple, si on n'aime pas le séparateur décimal, on peut demander sa suppression avec un appel à la commande `\npthousandsep`. Par exemple :

```
\npthousandsep{}\ucm{1.4142}
```

donne le résultat 1,4142 cm sans l'espace fine de séparation des milliers. Pour plus de détails, voir la documentation de l'extension `numprint`. La classe offre une grande quantité de commandes pour afficher un nombre avec une unité. Toutes ces commandes existent au format minuscule et majuscule comme le `\ucm` et `\Ucm` ci-dessus. On trouve :

Macros d'unités					
Unités de longueur					
<code>\uym{1234}</code>	1 234 ym	<code>\uzm{1234}</code>	1 234 zm	<code>\uam{1234}</code>	1 234 am
<code>\ufm{1234}</code>	1 234 fm	<code>\upm{1234}</code>	1 234 pm	<code>\unm{1234}</code>	1 234 nm
<code>\umicron{1234}</code>	1 234 μm	<code>\umm{1234}</code>	1 234 mm	<code>\ucm{1234}</code>	1 234 cm
<code>\udm{1234}</code>	1 234 dm	<code>\um{1234}</code>	1 234 m	<code>\udam{1234}</code>	1 234 dam
<code>\uhm{1234}</code>	1 234 hm	<code>\ukm{1234}</code>	1 234 km	<code>\uMm{1234}</code>	1 234 Mm
<code>\uGm{1234}</code>	1 234 Gm	<code>\uTm{1234}</code>	1 234 Tm	<code>\uPm{1234}</code>	1 234 Pm
<code>\uEm{1234}</code>	1 234 Em	<code>\uZm{1234}</code>	1 234 Zm	<code>\uYm{1234}</code>	1 234 Ym
Unités de surface					
<code>\ummq{1234}</code>	1 234 mm ²	<code>\ucmq{1234}</code>	1 234 cm ²	<code>\udmq{1234}</code>	1 234 dm ²
<code>\umq{1234}</code>	1 234 m ²	<code>\udamq{1234}</code>	1 234 dam ²	<code>\uhmq{1234}</code>	1 234 hm ²
<code>\ukmq{1234}</code>	1 234 km ²	<code>\uha{1234}</code>	1 234 ha		
Unités de volume					
<code>\ummc{1234}</code>	1 234 mm ³	<code>\ucmc{1234}</code>	1 234 cm ³	<code>\udmc{1234}</code>	1 234 dm ³
<code>\umc{1234}</code>	1 234 m ³	<code>\udamc{1234}</code>	1 234 dam ³	<code>\uhmc{1234}</code>	1 234 hm ³
<code>\ukmc{1234}</code>	1 234 km ³	<code>\uhl{1234}</code>	1 234 hl	<code>\udal{1234}</code>	1 234 dal
<code>\ul{1234}</code>	1 234 l	<code>\udl{1234}</code>	1 234 dl	<code>\ucl{1234}</code>	1 234 cl
<code>\uml{1234}</code>	1 234 ml	<code>\uהל{1234}</code>	1 234 hℓ	<code>\udael{1234}</code>	1 234 daℓ
<code>\uel{1234}</code>	1 234 ℓ	<code>\udel{1234}</code>	1 234 dℓ	<code>\uce1{1234}</code>	1 234 cℓ
<code>\umel{1234}</code>	1 234 mℓ				
Unités de temps					
<code>\us{1234}</code>	1 234 s	<code>\umin{1234}</code>	1 234 min	<code>\uh{1234}</code>	1 234 h
Unités de vitesse et débit					
<code>\ums{1234}</code>	1 234 m·s ⁻¹	<code>\umh{1234}</code>	1 234 m·h ⁻¹	<code>\ukms{1234}</code>	1 234 km·s ⁻¹
<code>\ukmin{1234}</code>	1 234 km·min ⁻¹	<code>\ukmh{1234}</code>	1 234 km·h ⁻¹	<code>\uls{1234}</code>	1 234 ℓ·s ⁻¹
Unités de masse					
<code>\umg{1234}</code>	1 234 mg	<code>\ucg{1234}</code>	1 234 cg	<code>\udg{1234}</code>	1 234 dg
<code>\ug{1234}</code>	1 234 g	<code>\udag{1234}</code>	1 234 dag	<code>\uhg{1234}</code>	1 234 hg
<code>\ukg{1234}</code>	1 234 kg	<code>\uT{1234}</code>	1 234 T		



Macros d'unités (suite)					
Unités diverses					
<code>\upc{1234}</code>	1 234 %	<code>\udeg{1234}</code>	1 234 °	<code>\udegc{1234}</code>	1 234 °C
<code>\ueuro{1234}</code>	1 234 €				

Si une unité semble manquer, on peut toujours en créer une autre avec la commande `\newunit`. Celle-ci demande deux arguments, le premier étant le fin du nom de la commande et le deuxième la façon de composer l'unité. Par exemple, la classe déclare `\ucm` avec la syntaxe :

```
\newunit{cm}{\text{cm}}
```

La commande `\text` est nécessaire parce que l'argument est composé en mode mathématique. Si on veut une commande pour exprimer une donnée en newtons, il suffit donc de faire :

```
\newunit{N}{\text{N}}  
\uN{9.81}
```

qui donne : 9,81 N

De même, si l'aspect d'une unité n'est pas celui souhaité, on peut le modifier en utilisant la commande `\renewunit`. Celle-ci fonctionne strictement comme `\newunit`. Par exemple :

```
Avant \ukmh{5}.  
\renewunit{kmh}{\text{km/h}}  
Après \ukmh{5}.
```

donne le résultat : Avant $5 \text{ km} \cdot \text{h}^{-1}$. Après 5 km/h.



Extention sesammanuelTIKZ.sty

Connaissances du collège nécessaires à ce chapitre

Sésamath ayant pour objectif de faciliter la diffusion libre de ses ressources, le code source du manuel *Sésamath* 2^{de} a été volontairement le plus neutre possible.

Mais pour une homogénéité du rendu de la version imprimable, des commandes personnalisées ont dû être implémentées.

Vous avez pu voir, par exemple, les tableaux dans le chapitre précédent.

► Ce chapitre traite de l'homogénéisation des figures.



Auto-évaluation

Des ressources numériques pour préparer le chapitre sur manuel.sesamath.net



- 1 Pourquoi avoir choisi de créer une extension à part de la classe ?
- 2 Pourquoi avoir choisi TIKZ et non pas PSTRIKS ?

▶▶▶ Voir solutions p. 74



ATTENTION : Cette extension n'est nécessaire que pour utiliser des sources du manuel *Sésamath* 2^{de} qui contiennent des figures.

1. L'appel à TIKZ

L'extension commence par appeler TIKZ et les bibliothèques nécessaires.

```
\usepackage{tikz} % appel général
\usepackage{pgf} % appel général
\usepackage{tkz-tab} % extension pour les tableaux de variations
\usetikzlibrary{arrows} % librairie pour les flèches
\usetikzlibrary{patterns} % librairie pour les hachures et les pointillés
```

2. La feuille de style

Le code suivant crée la feuille de style du manuel (épaisseur et couleur des traits).

```
\tikzstyle{general}=[line width=0.3mm, >=stealth, x=1cm, y=1cm, line cap=round, line join=round]
\definecolor{CyanTikz40}{cmyk}{.4,0,0,0}
\definecolor{CyanTikz20}{cmyk}{.2,0,0,0}
\tikzstyle{quadrillage}=[line width=0.3mm, color=CyanTikz40]
\tikzstyle{quadrillageNIV2}=[line width=0.3mm, color=CyanTikz20]
\tikzstyle{quadrillage55}=[line width=0.3mm, color=CyanTikz40, xstep=0.5, ystep=0.5]
\tikzstyle{cote}=[line width=0.3mm, <->]
\tikzstyle{epais}=[line width=0.5mm, line cap=butt]
\tikzstyle{tres epais}=[line width=0.8mm, line cap=butt]
\tikzstyle{axe}=[line width=0.3mm, ->, color=Noir, line cap=rect]
```

En détails :

SYNTAXE :

```
\tikzstyle{general}=[line width=0.3mm, >=stealth, x=1cm, y=1cm, line cap=round, line join=round]
```

définit pour tous les graphiques,

- l'épaisseur des traits (ici 0,3 mm correspondant à l'épaisseur minimum pour que les rotatives d'impression professionnelles fassent de la polychromie.)
- la forme de flèches en bout de segments (ici `stealth` correspond à une flèche en forme d'aile d'avion)
- l'unité choisie (ici le cm en abscisse et en ordonnée)
- la forme des extrémités de segments (ici `round` correspond à une forme arrondie)
- la forme des jonctions de segments (ici `round` correspond à une forme arrondie)





Ainsi, le code minimum pour une figure TIKZ sera :

```

\begin{tikzpicture}[general]
...
\end{tikzpicture}

```

REMARQUE : Le choix de l'unité en cm a été dicté pour faciliter les recherches des coordonnées des points lors de la construction des figures. La réduction ou l'agrandissement des figures se fait avec les options `scale`, `xscale` et `yscale`

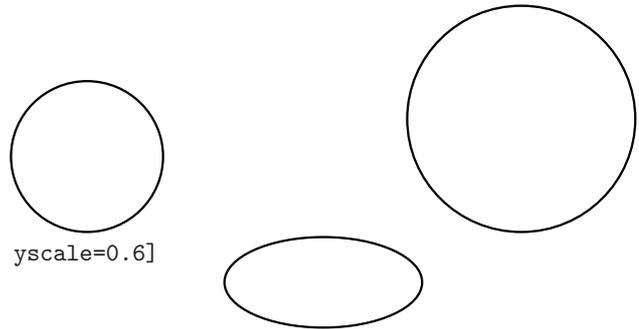
Le code

```

\begin{tikzpicture}[general]
\draw (0,0) circle (1);
\end{tikzpicture}
\begin{tikzpicture}[general, scale=1.5]
\draw (0,0) circle (1);
\end{tikzpicture}
\begin{tikzpicture}[general, xscale=1.3, yscale=0.6]
\draw (0,0) circle (1);
\end{tikzpicture}

```

donne le résultat :



SYNTAXE : Les commandes suivantes permettent de faire des quadrillages avec l'option `grid`

```

\definecolor{CyanTikz40}{cmyk}{.4,0,0,0}
\definecolor{CyanTikz20}{cmyk}{.2,0,0,0}
\tikzstyle{quadrillage}=[line width=0.3mm, color=CyanTikz40]
\tikzstyle{quadrillageNIV2}=[line width=0.3mm, color=CyanTikz20]
\tikzstyle{quadrillage55}=[line width=0.3mm, color=CyanTikz40, xstep=0.5, ystep=0.5]

```

L'épaisseur des traits ne pouvant être diminuer pour avoir deux niveaux de quadrillage, c'est la couleur du trait qui fait la différence. Le quadrillage de niveau 2 devra être fait avant le quadrillage simple pour que la couleur foncée recouvre la couleur claire.

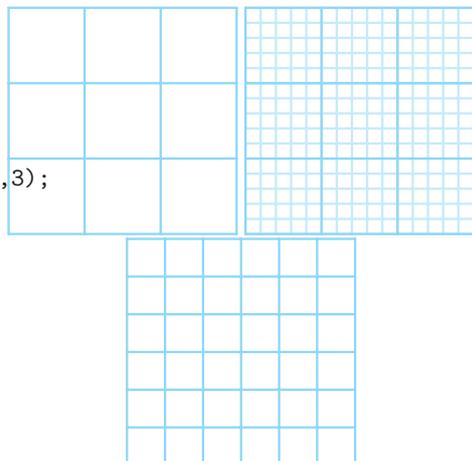
Le code

```

\begin{tikzpicture}[general]
\draw[quadrillage] (0,0) grid (3,3);
\end{tikzpicture}
\begin{tikzpicture}[general]
\draw[quadrillageNIV2, step=0.2] (0,0) grid (3,3);
\draw[quadrillage] (0,0) grid (3,3);
\end{tikzpicture}
\begin{tikzpicture}[general]
\draw[quadrillage55] (0,0) grid (3,3);
\end{tikzpicture}

```

donne le résultat :





SYNTAXE : Les commandes suivantes donnent différents types de segments (épaisseur et extrémités).

Lorsque l'épaisseur du trait est augmentée, la forme ronde de l'extrémité déborde. Les jonctions ne sont pas nettes. L'extrémité a donc été changée pour `butt` (coupe un peu avant) ou `rect` (coupe nette aux coordonnées).

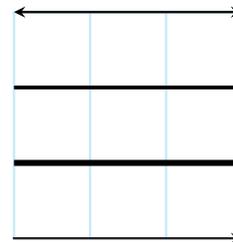
Les exemples ci-dessous sont donnés sur un quadrillage pour mieux apprécier la différence (le quadrillage est ou n'est pas recouvert).

```
\tikzstyle{cote}=[line width=0.3mm, <->]
\tikzstyle{epais}=[line width=0.5mm, line cap=butt]
\tikzstyle{tres epais}=[line width=0.8mm, line cap=butt]
\tikzstyle{axe}=[line width=0.3mm, ->, color=Noir, line cap=rect]
```

Le code

```
\begin{tikzpicture}[general]
\draw[quadrillageNIV2] (0,0) grid (3,3);
\draw[cote] (0,3)--(3,3);
\draw[epais] (0,2)--(3,2);
\draw[tres epais] (0,1)--(3,1);
\draw[axe] (0,0)--(3,0);
\end{tikzpicture}
```

donne le résultat :



3. Les commandes personnalisées

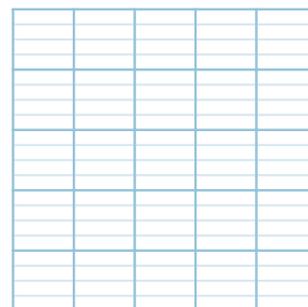
SYNTAXE : La commande suivante permet d'obtenir un quadrillage Seyes. Les arguments obligatoires sont les coordonnées du coin inférieur gauche et du coin supérieur droit. Les couleurs utilisées sont celles du nuancier de la classe.

```
\newcommand{\quadrillageSeyes}[2]{\draw[line width=0.3mm, color=A1!10, ystep=0.2, xstep=0.8]
#1 grid #2;
\draw[line width=0.3mm, color=A1!30, xstep=0.8, ystep=0.8] #1 grid #2; }
```

Le code

```
\begin{tikzpicture}[general]
\quadrillageSeyes{(0,0)}{(4,4)}
\end{tikzpicture}
```

donne le résultat :





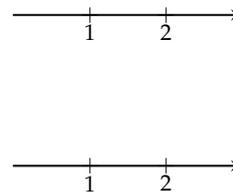
SYNTAXE : La commande suivante crée un axe des abscisses gradués pour un repère de type fonction. Les arguments obligatoires sont dans l'ordre : l'abscisse minimale, l'abscisse maximale, la liste des abscisses à graduer. Un argument optionnel permet de translater verticalement l'axe si les ordonnées ne commencent pas à 0.

```
\newcommand{\axeX}[4][0]{\draw[axe] (#2,#1)--(#3,#1); \foreach \x in {#4}
{\draw (\x,#1) node {\small $+$};
\draw (\x,#1) node[below] {\small $\x$};}}
```

Le code

```
\begin{tikzpicture}[general]
\axeX[2]{0}{3}{1, 2}
\axeX{0}{3}{1, 2}
\end{tikzpicture}
```

donne le résultat :



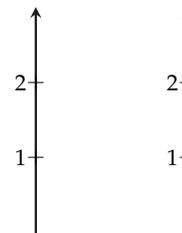
SYNTAXE : La commande suivante crée de même l'axe de ordonnées d'un repère de type fonction

```
\newcommand{\axeY}[4][0]{\draw[axe] (#1,#2)--(#1,#3); \foreach \y in {#4}
{\draw (#1, \y) node {\small $+$}; \draw (#1, \y) node[left] {\small $\y$};}}
```

Le code

```
\begin{tikzpicture}[general]
\axeY[2]{0}{3}{1,2}
\axeY{0}{3}{1,2}
\end{tikzpicture}
```

donne le résultat :

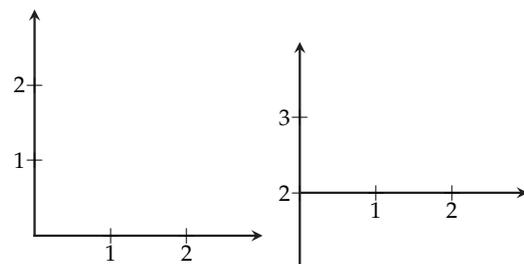


Soit, ensemble

Le code

```
\begin{tikzpicture}[general]
\axeY{0}{3}{1,2}
\axeX{0}{3}{1, 2}
\end{tikzpicture}
\begin{tikzpicture}[general]
\axeY[1]{4}{2,3}
\axeX[2]{0}{3}{1,2}
\end{tikzpicture}
```

donne le résultat :





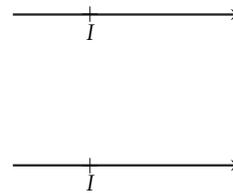
SYNTAXE : La commande suivante crée un axe des abscisses pour un repère de géométrie avec I pour repérer l'unité et sans graduation. Les arguments obligatoires sont dans l'ordre : l'abscisse minimale et l'abscisse maximale. Un argument optionnel permet de traduire verticalement l'axe si les ordonnées ne commencent pas à 0.

```
\newcommand{\axeOI}[3][0]{\draw[axe] (#2,#1)--(#3,#1);
\draw (1,#1) node {\small $$$};
\draw (1,#1) node[below] {\small $I$};}
```

Le code

```
\begin{tikzpicture}[general]
\axeOI[2]{0}{3}
\axeOI{0}{3}
\end{tikzpicture}
```

donne le résultat :



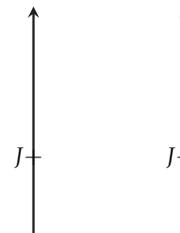
SYNTAXE : La commande suivante définit de même l'axe des ordonnées.

```
\newcommand{\axeOJ}[3][0]{\draw[axe] (#1,#2)--(#1,#3);
\draw (#1, 1) node {\small $$$};
\draw (#1, 1) node[left] {\small $J$};}
```

Le code

```
\begin{tikzpicture}[general]
\axeOJ[2]{0}{3}
\axeOJ{0}{3}
\end{tikzpicture}
```

donne le résultat :

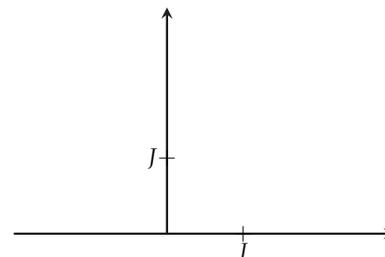


Soit, ensemble

Le code

```
\begin{tikzpicture}[general]
\axeOI{-2}{3}
\axeOJ{0}{3}
\end{tikzpicture}
```

donne le résultat :





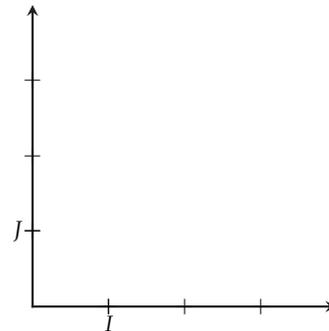
SYNTAXE : Les commandes suivantes permettent de rajouter des traits de graduation sur les axes créés avec les commandes `\axe0I` et `\axe0J`. Un argument obligatoire : la liste des abscisses ou ordonnées. Un argument optionnel : la position de l'axe si autre que 0 (voir commande `\axeX`).

```
\newcommand{\axeXgraduation}[2][0]{\foreach \x in {#2} {\draw (\x,#1) node {\small $+$};}}
\newcommand{\axeYgraduation}[2][0]{\foreach \y in {#2} {\draw (#1, \y) node {\small $+$}; }}
```

Le code

```
\begin{tikzpicture}[general]
\draw[axe] (0,0)--(4,0);
\axeXgraduation{1, 2, 3}
\end{tikzpicture}
```

donne le résultat :



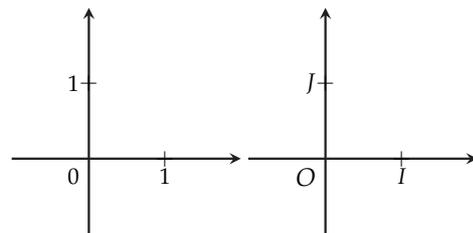
SYNTAXE : Les commandes suivantes permettent d'afficher l'origine dans les deux types de repères : 0 pour les types fonction et O pour les géométriques. Attention, suivant le redimensionnement choisi, la position de l'origine peut empêcher de placer une abscisse et/ou une ordonnée -1 .

```
\newcommand{\origine}{\draw (0,0) node[below left] {\small $0$};}
\newcommand{\origine0}{\draw (0,0) node[below left] {$0$};}
```

Le code

```
\begin{tikzpicture}[general]
\axeX{-1}{2}
\axeY{-1}{2}
\origine
\end{tikzpicture}
\begin{tikzpicture}[general]
\axe0I{-1}{2}
\axe0J{-1}{2}
\origine0
\end{tikzpicture}
```

donne le résultat :





SYNTAXE : Les commandes suivantes permettent de placer le nom d'un point à partir de ses coordonnées.

- la commande `\point` donnera juste le nom du point
- la commande `\pointGraphique` donnera le nom du point et repérera la point par une croix en forme de +
- la commande `\pointFigure` donnera le nom du point et repérera la point par une croix en forme de ×

```
\newcommand{\point}[4]{\draw (#1,#2) node[#4] {$#3$};}
\newcommand{\pointGraphique}[4]{\draw (#1,#2) node[#4] {$#3$};}
\draw (#1,#2) node {$+$};}
\newcommand{\pointFigure}[4]{\draw (#1,#2) node[#4] {$#3$};}
\draw (#1,#2) node {$\times$};}
```

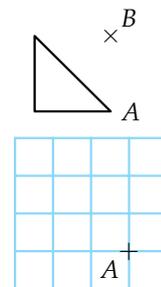
À chaque fois, quatre arguments obligatoires :

- abscisse du point
- ordonnée du point
- nom du point
- positionnement du nom par rapport au point, au choix : `above` (dessus) `below` (dessous) `right` (droite) `left` (gauche) (on peut combiner : `below left` positionnement vertical suivi de l'horizontal)

Le code

```
\begin{tikzpicture}[general]
\draw (0,0)--(1,0)--(0,1)--cycle;
\point{1}{0}{A}{right}
\pointFigure{1}{1}{B}{above right}
\end{tikzpicture}
\begin{tikzpicture}[general]
\draw[quadrillage55] (0,0)grid(2,2);
\pointGraphique{1.5}{0.5}{A}{below left}
\end{tikzpicture}
```

donne le résultat :



SYNTAXE : Dans des figures complexes, on peut être amené à définir des points qui servent régulièrement avec la commande `\coordinate`. Les commandes suivantes fonctionnent comme les précédentes mais les deux premiers arguments demandant l'abscisse et l'ordonnée du point sont remplacées par un argument unique qui est le code du point. Généralement, il s'agit aussi du nom du point

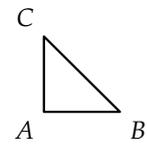
```
\newcommand{\pointC}[3]{\draw (#1) node[#3] {$#2$};}
\newcommand{\pointCGraphique}[3]{\draw (#1) node[#3] {$#2$};}
\draw (#1) node {$+$};}
\newcommand{\pointCFigure}[3]{\draw (#1) node[#3] {$#2$};}
\draw (#1) node {$\times$};}
```



Le code

```
\begin{tikzpicture}[general]
\coordinate (A) at (0,0);
\coordinate (B) at (1,0);
\coordinate (C) at (0,1);
\draw (A)--(B)--(C)--cycle;
\pointC{A}{A}{below left}
\pointC{C}{C}{above left}
\pointC{B}{B}{below right}
\end{tikzpicture}
```

donne le résultat :



SOLUTIONS

Chapitre M1

Sesamanuel Documentation

Chapitre M1

Introduction

Auto-évaluation

- 1 S'il ne doit y avoir qu'une seule section à lire dans le premier chapitre, c'est celle traitant du mode de compilation (section 3).
- 2 L'utilisation de la classe pour composer un seul élément ou quelques éléments isolés est décrite dans le chapitre 3.
- 3 La classe a été pensée pour être hautement configurable. S'il y a des aspects fondamentaux qui ne vous conviennent pas, c'est peut-être que cette classe ne vous convient pas !
- 4 La classe et l'extension sont des projets libres. On peut donc modifier ce qu'on veut. Pour éviter toute erreur de manipulation, il est cependant plus prudent de modifier aussi le nom de la classe. N'oubliez pas de modifier la ligne `\ProvidesClass{sesamanuel}` de la classe ou la ligne `\ProvidesPackage{sesamanuel}` de l'extension.

Chapitre M2

Classe pour un ouvrage

Auto-évaluation

- 1 Soit vous disposez d'une capacité hors norme pour la vitesse de lecture, soit il ne faut pas trop se faire d'illusion et il est impossible d'assimiler l'ensemble des fonctionnalités en moins de deux heures.

- 2 D'après ce qui a été dit à la page qui suit cet exercice, il suffit de placer, avant l'appel à l'environnement `autoeval` le code suivant :

```
\renewcommand*\StringAE{%  
  Exercices pour auteurs  
}
```

- 3 Parce que si on a le source sous les yeux, on a également la solution de l'exercice sous les yeux !

S'entraîner

- 2 On sait que $1 = 1$ avec l'exercice précédent donc $1 + 1 = 1 + 1$, c'est-à-dire $2 = 2$.

Auto-évaluation QCM

- 4 a
- 5 c

Chapitre M3

Classe pour quelques feuilles

Auto-évaluation

- 1 Les codes sources des manuels peuvent être téléchargés
 - par chapitre (fichiers zip) sur le site <http://www.sesamath.net/>. Cliquez sur Les manuels puis sur TÉLÉCHARGEMENTS. Actuellement, seuls les manuels de lycée utilisent \LaTeX et cette classe.
 - par vignette du manuel numérique en cliquant sur le 
- 2 Comme `beamer` est une classe, la seule solution est d'utiliser l'extension `sesacah.sty`.

Chapitre M4

Détails typographiques

Auto-évaluation

- 1 Normalement, on a une erreur de compilation. On peut alors lire le rapport de compilation ou lire la section qui suit pour comprendre ce qui s'est passé.
- 2 On ne peut pas.
- 3 La possibilité de modifier le comportement de la classe, en particulier les textes fixes, devrait permettre de faire une sorte de `sesaphysique` à la place d'un `sesamath`.

Chapitre M5

Extension

`sesamanuelTIKZ.sty`

Auto-évaluation

- 1 Les commandes spéciales TIKZ n'ont pas été incluses dans la classe pour permettre à l'utilisateur de rester libre de la mise en forme de ses figures. Le niveau d'exigence pour une impression en rotatives n'est pas nécessaire à tous.
- 2 Citons le vénérable Sage « le meilleur logiciel est celui qu'on a l'habitude d'utiliser »... Une grande majorité des auteurs du manuel *Sésamath* 2^{de} étant utilisateurs de TIKZ, le choix a donc été fait d'utiliser TIKZ pour le manuel même si cet échantillon n'est peut-être pas représentatif de la communauté mondiale des utilisateurs de \LaTeX .

