

# **style-tex3R**

Vincent Crombez & Frédéric Léothaud

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Créer un nouveau style</b>	<b>4</b>
<b>2 Principe de modification</b>	<b>4</b>
<b>3 FormatUtilisateur()</b>	<b>5</b>
3.1 Listes et énumérations . . . . .	5
3.2 Chapitre, partie et sous-partie . . . . .	5
3.3 if FORMAT == "fiche" . . . . .	6
3.4 if FORMAT == "diapo" . . . . .	7
<b>4 PoliceUtilisateur()</b>	<b>9</b>
<b>5 CouleursUtilisateur()</b>	<b>9</b>
5.1 if PRINT . . . . .	9
5.2 else . . . . .	10
<b>6 LogosUtilisateur()</b>	<b>10</b>
<b>7 StylesTitresUtilisateur()</b>	<b>11</b>
7.1 Définition des styles . . . . .	11
7.2 if FORMAT == "fiche" . . . . .	11
7.3 if FORMAT == "diapo" . . . . .	12
<b>8 TitresUtilisateur()</b>	<b>12</b>
8.1 Formatage des titres . . . . .	12
8.2 Formatage de la TOC . . . . .	13
<b>9 EnvironnementsUtilisateur()</b>	<b>13</b>
9.1 Environnements de cours . . . . .	13
9.2 Environnement enonce . . . . .	13
9.3 Environnement correction . . . . .	14
9.4 Formatage de la TOC . . . . .	14
<b>10 mesParametres(str)</b>	<b>14</b>
<b>11 NiveauUtilisateur(arg)</b>	<b>15</b>

# Introduction

La `style-tex3R` fait partie d'un projet comprenant :

- Une classe, un style et des templates de documents pratiques
- Un environnement LaTeX portable
- Une extension (basée sur `latex-workshop`)

Il est préférable d'aller d'abord lire la documentation de `classe-tex3R` qui explique l'ensemble des commandes disponibles.

L'objectif de ce guide est de guider la personnalisation du style à appliquer aux environnements et commandes de la `classe-tex3R`, afin d'avoir une apparence propre selon les goûts de chacun, tout en gardant la possibilité de transférer son document à un autre utilisateur de la `classe-tex3R`, qui lui le compilera avec le style de son choix.

# 1 Créer un nouveau style

Afin d'utiliser son style personnalisé, il suffit de créer une copie du fichier `style-tex3R.sty` dans le même dossier, de le renommer `monstyle.sty` et de travailler sur ce dernier. Le fichier `style-tex3R` se trouve dans l'emplacement suivant :

```
TeX3R-ClasseStyle
├── documentation
├── templates
├── tex
│   ├── fonts
│   ├── lualatex
│   │   └── tex3R
│   │       ├── classe-tex3R.cls
│   │       └── style-tex3R.sty
├── .gitignore
└── README.md
```

Une fois le fichier `monstyle.sty` créé il est nécessaire d'ouvrir MiKTeX Console et de faire Tasks > Refresh file name database.

Ensuite, pour appeler `monstyle` dans un nouveau document, il suffit de changer le préambule comme suit :

```
\documentclass{classe-tex3R}
\usepackage{monstyle}
\parametrage

\begin{document}

%Le texte du document

\end{document}
```

## 2 Principe de modification

`classe-tex3R` et `style-tex3R` sont codés majoritairement en `lua`. L'idée derrière cela est d'avoir une part minimale de choses à modifier dans le style, dans des chaînes de caractères `lua`, qui seront ensuite insérées et utilisées par les commandes de la classe. Ceci étant, les choses pouvant être modifiées sont toujours entre `[[ ]]` ou entre `' '` qui sont les délimiteurs des chaînes de caractères en `lua`.

Cette expérience simplifiée évite d'avoir à aller voir dans de nombreuses documentations pour connaître la syntaxe exacte pour chaque paramètre.

Par la suite, nous détaillerons chacune des fonctions `lua`, et à quoi correspondent les différentes parties modifiables.

## 3 FormatUtilisateur()

## 3.1 Listes et énumérations

```
PAGE.enumi = [[ \textbf{\arabic*.\~} ]]
PAGE.enumii = [[ \textbf{\alph*.\~} ]]
PAGE.itemi = [[ \textbullet ]]
PAGE.itemii = [[ $\circ$ ]]
```

- enumi : liste numérotée de niveau 1.
- enumii : liste numérotée de niveau 2.

\Alph\*, \alph\*, \roman\*, \Roman\*, \arabic\*

- itemi : liste à puces de niveau 1.
- itemii : liste à puces de niveau 2.

Possibilité de mettre n'importe quel symbole.

## 3.2 Chapitre, partie et sous-partie

```
PAGE.sectionname = [[ \LARGE\bfseries\textcolor{couleursection} ]]
PAGE.subsectionname = [[ \Large\bfseries\textcolor{couleursubsection} ]]
PAGE.subsubsectionname = [[ \normalfont\large\bfseries\textcolor{couleursubsubsection} ]]
```

Format du texte des \section, \subsection et \subsubsection. Pour rappel, les \section ne sont pas utilisées car remplacées par les titres. Les \subsection et \subsubsection correspondent aux commandes \partie et \souspartie

```
PAGE.stylesections = [[ \colorbox{#1}{\textcolor{white}{\normalfont#2\textbf{{#3}}}} ]]

PAGE.sectionnumber = [[ \stylesections{couleursection}{\LARGE}{~\Roman{section}~}\, ]]
PAGE.subsectionnumber = [[ \stylesections{couleursubsection}{\LARGE}{\Alph{subsection}}\, ]]
PAGE.subsubsectionnumber = [[ \stylesections{couleursubsubsection}{\LARGE}{\arabic{subsubsection}}\, ]]
```

- stylesections : style du numéro devant les sections, subsections et subsubsections (par défaut un carré de couleur avec le nombre ou la lettre en blanc). Trois argument :
  - #1 = couleur
  - #2 = Taille du texte
  - #3 = Style de la numérotation (Roman, arabic, etc)
- sectionnumber : style du numéro des \section
- subsectionnumber : style du numéro des \subsection
- subsubsectionnumber : style du numéro des \subsubsection

```
PAGE.sectionsformat = [[ #3 \underline{#4}\par\vspace{0.5em} ]]
```

sectionsformat : formatage global des \section, \subsection et \subsubsection. Par défaut, texte souligné sans espacement horizontal.

- #3 = emplacement de la numérotation
- #4 = emplacement du texte

```

PAGE.chapitrenormalformat = [[ \huge\bfseries Chapitre \thepart ]]
PAGE.chapitreetoileformat = [[]]
PAGE.partformat = [[
  \newpage
  \hfill
  \vfill
  \begin{tcolorbox}[
    colframe = black,
    colback=white,
    coltitle=white,
    sharp corners,
    valign=center,
    top=2cm,
    bottom=2cm,
    left=0.5cm,
    enhanced,
    before skip=0cm,
  ]
  \formatchapitre
  \tcblower
  {\huge\bfseries #3}
  \end{tcolorbox}
  \hfill
  \vfill
]]

```

- chapitrenormalformat : contenu de \formatchapitre pour la commande \chapitre
- chapitreetoileformat : contenu de \formatchapitre pour la commande \chapitre\*
- partformat : formatage de la \part, ce qui formate également la commande \chapitre

Rappel : la commande \chapitre ne sert que pour les méga-documents, page de séparation entre deux chapitres (par défaut, grand cadre dans une tcolorbox, voir le code ci-dessous)

### 3.3 if FORMAT == "fiche"

```

PAGE.width = '21cm'
PAGE.height = '29.7cm'
PAGE.parindent = '0pt'
PAGE.parskip = '0pt'

PAGE.headertop = '1cm'
PAGE.headerbottom = '0.3cm'
PAGE.headerright = '1cm'
PAGE.headerleft = '1cm'
PAGE.headermarginparsep = '0cm'
PAGE.headermarginparwidth = '0cm'
PAGE.headerheadheight = '1cm'
PAGE.headerheadsep = '0.5cm'
PAGE.headerfootskip = '0.7cm'

PAGE.top = '1cm'
PAGE.bottom = '0.3cm'
PAGE.right = '1cm'

```

```

PAGE.left = '1cm'
PAGE.marginparsep = '0cm'
PAGE.marginparwidth = '0cm'
PAGE.headheight = '0cm'
PAGE.headsep = '0cm'
PAGE.footskip = '0.7cm'

```

### 3.4 if FORMAT == "diapo"

```

PAGE.width = '28.8cm'
PAGE.height = '18cm'
PAGE.parindent = '0pt'
PAGE.parskip = '0pt'

PAGE.headertop = '0.5cm'
PAGE.headerbottom = '2cm'
PAGE.headerright = '1cm'
PAGE.headerleft = '1cm'
PAGE.headermarginparsep = '0cm'
PAGE.headermarginparwidth = '0cm'
PAGE.headerheadheight = '1.5cm'
PAGE.headerheadsep = '0.5cm'
PAGE.headerfootskip = '1cm'

PAGE.top = '0.5cm'
PAGE.bottom = '2cm'
PAGE.right = '1cm'
PAGE.left = '1cm'
PAGE.marginparsep = '0cm'
PAGE.marginparwidth = '0cm'
PAGE.headheight = '0cm'
PAGE.headsep = '0cm'
PAGE.footskip = '1cm'

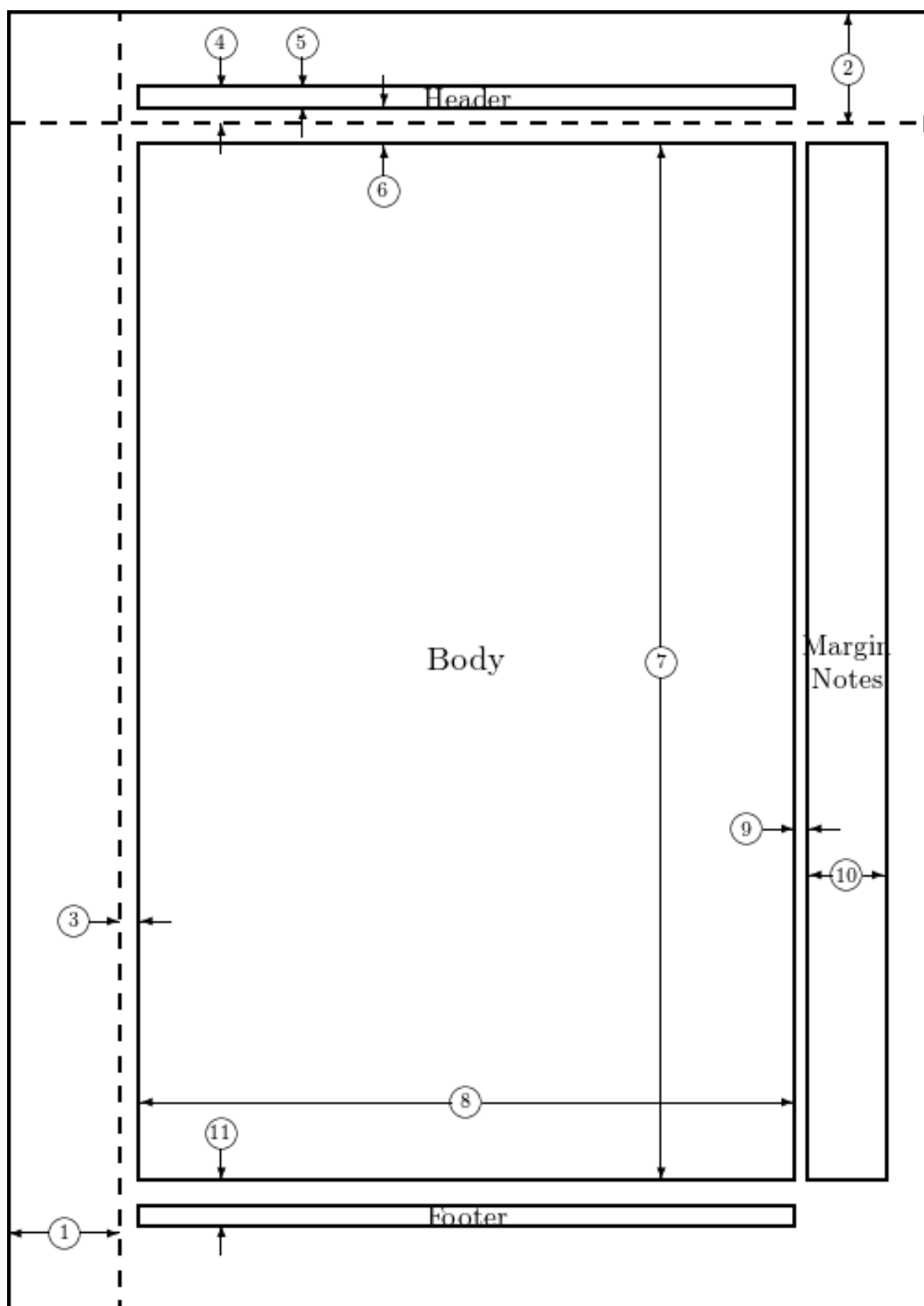
```

Paramétrage des marges et des dimensions du document pour le format `fiche` et pour le format `diapo`, avec ou sans header (le header est utilisé lorsque l'option `Titre` est `true`, afin d'avoir un titre persistant).

- `width` : largeur de la page
- `height` : hauteur de la page
- `parindent` : taille de l'indentation en début de paragraphe
- `parskip` : taille de l'écart entre deux paragraphes

Les autres variables renvoient pour le document avec ou sans header (selon la variable `Titre`) aux dimensions usuelles, visibles sur le document en page suivante.

Lors de tests sur la disposition de la page, il est recommandé d'utiliser `\usepackage{showframe}` dans le préambule, qui permet de visualiser par des cadres les marges du document.



- |    |                       |    |                                  |
|----|-----------------------|----|----------------------------------|
| 1  | one inch + \hoffset   | 2  | one inch + \voffset              |
| 3  | \oddsidemargin = 13pt | 4  | \topmargin = -23pt               |
| 5  | \headheight = 12pt    | 6  | \headsep = 25pt                  |
| 7  | \textheight = 674pt   | 8  | \textwidth = 426pt               |
| 9  | \marginparsep = 10pt  | 10 | \marginparwidth = 50pt           |
| 11 | \footskip = 30pt      |    | \marginparpush = 5pt (not shown) |
|    | \hoffset = 0pt        |    | \voffset = 0pt                   |
|    | \paperwidth = 597pt   |    | \paperheight = 845pt             |



## 4 PoliceUtilisateur()

```
POLICE.taillefiche = '12pt'  
POLICE.taillédiapo = '20pt'  
POLICE.tableau = '1.4'  
POLICE.texte = 'Calibri'  
POLICE.math = 'STIX Two Math'  
POLICE.mathcal = 'Libertinus Math Regular'  
POLICE.important = [[ \hl{#1} ]]
```

- `taillefiche` : taille de police par défaut pour les documents au format `fiche`
- `taillédiapo` : taille de police par défaut pour les documents au format `diapo`
- `tableau` : écartement naturel des lignes de tableaux (par défaut 140%)
- `texte` : police par défaut pour le texte
- `math` : police par défaut pour le math mode
- `mathcal` : police par défaut pour les lettres calligraphiées en math mode
- `important` : style pour la commande `\important` (par défaut, surligné)

Si les polices ne sont pas installées sur le système, la classe `tex3R` procédera à une substitution par une police par défaut de  $\TeX$ . Attention, les polices mathématiques sont des polices spéciales. A voir : [The  \$\TeX\$ Font Catalogue](#)

## 5 CouleursUtilisateur()

### 5.1 if PRINT

```
COULEURS.ligne = 'black'  
COULEURS.interligne = 'black!30'  
COULEURS.section = 'black'  
COULEURS.subsection = 'black!70'  
COULEURS.subsubsection = 'black!40'  
COULEURS.carreau = 'black!30'  
COULEURS.surligneligne = 'black!20'  
COULEURS.surligneligne = 'white'  
COULEURS.boitecours = 'white'  
COULEURS.background = 'white'  
  
COULEURS.application = 'black'  
COULEURS.definition = 'black'  
COULEURS.convention = 'black'  
COULEURS.exercice = 'black'  
COULEURS.exemple = 'black'  
COULEURS.methode = 'black'  
COULEURS.preuve = 'black'  
COULEURS.propriete = 'black'  
COULEURS.remarque = 'black'  
COULEURS.enonce = 'black'  
COULEURS.correction = 'black'  
COULEURS.surligne = 'black!5'
```

```
COULEURS.surlignevisible = 'black!10'

COULEURS.rouge = 'black'
COULEURS.bleu = 'black'
COULEURS.vert = 'black'
```

## 5.2 else

```
COULEURS.ligne = 'SlateBlue'
COULEURS.interligne = 'LightSteelBlue'
COULEURS.section = 'black'
COULEURS.subsection = 'Red'
COULEURS.subsubsection = 'Green'
COULEURS.carreau = 'blue!25'
COULEURS.surligneligne = 'blue!20'
COULEURS.surligneligne = 'yellow!20'
COULEURS.boitecours = 'black!5'
COULEURS.background = 'white'

COULEURS.application = 'black'
COULEURS.definition = 'Green'
COULEURS.convention = 'black'
COULEURS.exercice = 'Blue'
COULEURS.exemple = 'Blue'
COULEURS.methode = 'black'
COULEURS.preuve = 'black'
COULEURS.propriete = 'Red'
COULEURS.remarque = 'black'
COULEURS.enonce = 'black'
COULEURS.correction = 'black'
COULEURS.surligne = 'yellow'
COULEURS.surlignevisible = 'black!10'

COULEURS.rouge = 'Red'
COULEURS.bleu = 'Blue'
COULEURS.vert = 'Green'
```

Toutes les couleurs sont définies dans la classe, il n'est pas possible d'en rajouter dans le `lua`. L'intérêt de ces couleurs, est de pouvoir changer leur valeur en fonction du paramètre `Impression`. Le nom des couleurs en  $\text{\LaTeX}$  est couleur + variable (par exemple : couleursection ou couleurapplication)

- ligne; interligne : couleurs pour la commande `\lignes{seyes}`
- carreau : couleur pour la commande `\lignes{carreau}`
- surligne; surligne : couleurs pour la commande `\lignes{surligne}`
- boitecours : couleur de l'arrière plan des boîte de cours
- background : couleur de l'arrière plan des documents
- surligne : couleur de la commande `\hl` utilisée par la commande `\important`
- rouge; bleu; vert : couleurs du panel

## 6 LogosUtilisateur()

```
LOGOS.activite = [[ \reflectbox{\faPencil*} ]]
LOGOS.basique = ''
LOGOS.bilan = [[ \faEdit[regular] ]]
LOGOS.corrige = [[ \faEdit[regular] ]]
LOGOS.cours = [[ \faBook ]]
LOGOS.DM = [[ \faHome ]]
LOGOS.DS = [[ \faFile*[regular] ]]
LOGOS.flash = [[ \faUserSecret ]]
LOGOS.interro = [[ \faFile*[regular] ]]
LOGOS.TD = [[ \faEdit[regular] ]]
```

Logos utilisés pour les titres, voir le package fontawesome5. Le logo est stocké dans \logoactif, qui change en fonction du Type.

## 7 StylesTitresUtilisateur()

Les titres sont définis en trois temps :

1. Le style global, commun à tous les titres (par défaut, un logo, et un contenu souligné sur toute la longueur)
2. Le style particulier au Type du document (le nom du chapitre pour les fiches de TD ou faire apparaître NOM et Prénom pour les DS par exemple)
3. Le mot clé à faire apparaître au bon endroit pour les titres partageant le même style (DM, interro ou DS par exemple)

### 7.1 Définition des styles

```
STYLESTITRES.esacement = '0.5cm'
STYLESTITRES.titre = [[ {\bfseries\LARGE\logoactif~\underLine{#1}\par} ]]
STYLESTITRES['sans'] = [[ ]]
STYLESTITRES['un'] = [[ \titre{#1\hfill\mdseries\large\contenuniveau} ]]
STYLESTITRES['deux'] = [[ \titre{#1~\mdseries|~\contenuchapitre\hfill\large\contenuniveau} ]]
STYLESTITRES['trois'] = [[ \titre{#1~\mdseries|~{\large\mdseries NOM :} \hfill {\large\mdseries Prénom :} \hfill \mdseries\large\contenuniveau} ]]
```

- esacement : saut de ligne après le titre de début de page (à faire correspondre avec PAGE.headerheadsep)
- titre : définit le style global à tous les titres, comme expliqué précédemment

Pour définir des styles particuliers de titre, les commandes \contenuniveau, \contenuchapitre (donnant le contenu des options Niveau et Chapitre) sont accessibles.

il est possible de créer d'autres styles de particuliers de titre dans le lua sur le même modèle, au besoin.

### 7.2 if FORMAT == "fiche"

```
STYLESTITRES.activite = STYLESTITRES['un']
STYLESTITRES.basique = STYLESTITRES['sans']
```

```

STYLESTITRES.bilan = STYLESTITRES['un']
STYLESTITRES.corrige = STYLESTITRES['deux']
STYLESTITRES.cours = STYLESTITRES['deux']
STYLESTITRES.DM = STYLESTITRES['trois']
STYLESTITRES.DS = STYLESTITRES['trois']
STYLESTITRES.flash = STYLESTITRES['un']
STYLESTITRES.interro = STYLESTITRES['trois']
STYLESTITRES.TD = STYLESTITRES['deux']

```

## 7.3 if FORMAT == "diapo"

```

STYLESTITRES.activite = STYLESTITRES['un']
STYLESTITRES.basique = STYLESTITRES['sans']
STYLESTITRES.bilan = STYLESTITRES['un']
STYLESTITRES.corrige = STYLESTITRES['deux']
STYLESTITRES.cours = STYLESTITRES['deux']
STYLESTITRES.DM = STYLESTITRES['trois']
STYLESTITRES.DS = STYLESTITRES['trois']
STYLESTITRES.flash = STYLESTITRES['un']
STYLESTITRES.interro = STYLESTITRES['trois']
STYLESTITRES.TD = STYLESTITRES['deux']

```

Il est possible de changer le style du titre selon que l'on est en format `fiche` ou en format `diapo`.

# 8 TitresUtilisateur()

## 8.1 Formatage des titres

```

TITRES.format = [[ Activité ]]

TITRES.format = [[ ]]

TITRES.format = [[ Bilan ]]

TITRES.format = [[ Corrigé ]]

TITRES.format = [[ Chapitre~\thepart ]]

TITRES.format = [[ DM n°\stepcounter{compteurDM}\thecompteurDM ]]

TITRES.format = [[ DS n°\stepcounter{compteurDS}\thecompteurDS ]]

TITRES.format = [[ Questions Flash n°\stepcounter{compteurflash}\thecompteurflash ]]

TITRES.format = [[ Interrogation n°\stepcounter{compteurinterro}\thecompteurinterro ]]

TITRES.format = [[ TD ]]

```

`format` : permet de placer le mot clé dans l'emplacement du #1 des `STYLESTITRES['X']`

## 8.2 Formatage de la TOC

```
TITRES.tocactivite = 'Activité'
TITRES.tocbasique = 'Sans titre'
TITRES.tocbilan = 'Bilan'
TITRES.toccorrige = 'Corrigé'
TITRES.toccours = 'Cours'
TITRES.tocDM = 'DM'
TITRES.tocDS = 'DS'
TITRES.tocflash = 'Flash'
TITRES.tocinterro = 'Interrogation'
TITRES.tocTD = 'TD'
```

Mot-clés apparaissant selon le Type de document choisi dans les signets du document.

## 9 EnvironnementsUtilisateur()

### 9.1 Environnements de cours

```
ENVIRONNEMENTS.formatcours = [[
  \begin{tcolorbox}[
    colback=couleurboitecours,
    colframe=couleurbackground,
    boxrule=0pt,
    left=0.5em,
    top=0pt,
    bottom=0pt,
    right=0pt,
    enhanced,
    shield externalize=true,
    borderline west={4pt}{0pt}{#2}
  ]
  \textcolor{#2}{\large\underLine{#1} :}\par
  \medskip
  \BODY
\end{tcolorbox}
]]
```

### 9.2 Environnement enonce

```
ENVIRONNEMENTS.enonce = [[
  \iftheme
    \hfill{\itshape\footnotesize\contenutheme}\par\vspace{0.2em}{\color{couleurenonce}\hrule
      height 2pt}\par
  \fi
  \colorbox{couleurenonce}{\textcolor{white}{
    {
```

```

\Large\textbf{\thecompteurexercice}}
\ifdifficulte{~\raisebox{0.2em}{\scriptsize\contenudifficulte}}\fi
}
}
\ifcompetence\hspace{0.3em}\textcolor{couleurenonce}{\underline{\large\raisebox{0.1em}{\
contenucompetence}}}\fi\ifbareme\hfill {\bfseries/\contenubareme~points}\fi\par
\medskip
\BODY
\ifsource\par
\vspace{0.2em}{\color{couleurenonce}\hrule height 2pt}\par\vspace{0.2em}\hfill{\itshape\
footnotesize\contenusource}
\fi
]]

```

## 9.3 Environnement correction

```

ENVIRONNEMENTS.correction = [[
\colorbox{couleurcorrection}{\textcolor{white}{
\normalfont\Large\textbf{\thecompteurexercice} | Correction}
}\par
\medskip
\BODY
]]

```

## 9.4 Formatage de la TOC

```

ENVIRONNEMENTS.tocexercice = 'Exercice'
ENVIRONNEMENTS.tocenonce = 'Énoncé'
ENVIRONNEMENTS.toccorrection = 'Correction'

```

# 10 mesParametres(str)

```

function mesParametres(str)

if str == 'activite' or str == 'Activite' then
Type = 'activite'
Impression = true
Competence = true
Enonce = true
Stretch = true
elseif str == 'basique' or str == 'Basique' then

elseif str == 'bilan' or str == 'Bilan' then
Type = 'bilan'
Impression = true
Competence = true
Enonce = true

```

```

    Stretch = true
elseif str == 'corrige' or str == 'Corrige' then

elseif str == 'cours' or str == 'Cours' then
    Type = 'cours'
    Enonce = true
    Correction = true
    Stretch = false
elseif str == 'DM' then
    Type = 'DM'
    Impression = true
    Competence = true
    Enonce = true
    Stretch = true
elseif str == 'DS' then
    Type = 'DS'
    Impression = true
    Competence = true
    Enonce = true
    Stretch = true
    Bareme = true
elseif str == 'flash' or str == 'Flash' then

elseif str == 'interro' or str == 'Interro' then
    Type = 'interro'
    Impression = true
    Competence = true
    Enonce = true
    Stretch = true
    Bareme = true
elseif str == 'TD' then
    Type = 'TD'
    Header = true
    Impression = true
    Competence = true
    Enonce = true
    Stretch = true
end
end
end

```

## 11 NiveauUtilisateur(arg)

```

function NiveauUtilisateur(arg)
    local str = nil
    if arg == '6' then
        str = '6$^\\text{ème}$'
    elseif arg == '5' then
        str = '5$^\\text{ème}$'
    elseif arg == '4' then
        str = '4$^\\text{ème}$'
    elseif arg == '3' then
        str = '3$^\\text{ème}$'
    end
    NiveauDocument(str)
end

```

end