

# Petit guide : l'environnement exocbhlb

Boîtes d'exercices avec niveau de difficulté, type et couleurs extensibles

C. Pierquet

Juin 2026

**11**

FACILE ●○○○

Ut purus elit, vestibulum ut,  
placerat ac, adipiscing vitae,  
felis.

**12**

MOYEN ●●○○

Ut purus elit, vestibulum ut,  
placerat ac, adipiscing vitae,  
felis.

**13**

QCM

DIFFICILE ●●●○

Ut purus elit, vestibulum ut,  
placerat ac, adipiscing vitae,  
felis.

## Table des matières

|          |                                                      |           |
|----------|------------------------------------------------------|-----------|
| <b>1</b> | <b>Présentation</b>                                  | <b>2</b>  |
| <b>2</b> | <b>Analyse et explication du code source</b>         | <b>2</b>  |
| 2.1      | Dépendances                                          | 2         |
| 2.2      | Variables et trousseau de paramètres globaux         | 2         |
| 2.3      | Couleurs                                             | 3         |
| 2.4      | Pastilles de niveau (pifont)                         | 4         |
| 2.5      | Style de boîte tcolorbox                             | 4         |
| 2.6      | La couleur de sous-titre extensible : \cbhlbtextcoul | 5         |
| 2.7      | Sous-titres prédéfinis (type d'exercice)             | 5         |
| 2.8      | Génération du sous-titre                             | 6         |
| 2.9      | Construction du titre                                | 6         |
| 2.10     | Sélection du niveau (1 à 4, ou 0 pour le cours)      | 7         |
| <b>3</b> | <b>L'environnement exocbhlb</b>                      | <b>8</b>  |
| 3.1      | La macro de création de l'environnement              | 8         |
| 3.2      | Un compteur dédié : \CreationExoStyleH               | 8         |
| 3.3      | Mise à jour des paramètres globaux                   | 9         |
| <b>4</b> | <b>Utilisation</b>                                   | <b>10</b> |
| 4.1      | Paramétrage global                                   | 10        |
| 4.2      | Syntaxe de base                                      | 10        |
| 4.2.1    | Les cinq niveaux, sans type                          | 10        |
| 4.2.2    | Types prédéfinis (avec icône)                        | 11        |
| 4.2.3    | Type libre (texte brut ou texte formaté)             | 12        |
| 4.3      | Désactiver toutes les couleurs                       | 12        |
| 4.4      | Couleurs dédiées par type : couleur auto=false       | 12        |
| 4.5      | Ajouter un nouveau type                              | 13        |
| 4.5.1    | Juste une couleur dédiée (sans icône)                | 13        |
| 4.5.2    | Juste une icône (suit la couleur du niveau)          | 14        |
| 4.5.3    | Icône + couleur dédiée                               | 14        |
| 4.5.4    | Macro de génération automatique                      | 14        |
| 4.5.5    | Consignes d'utilisation                              | 15        |
| <b>5</b> | <b>Récapitulatif</b>                                 | <b>16</b> |

# 1 Présentation

exocbhlb est un environnement  $\text{\LaTeX}$  (basé sur `tcolorbox`), désormais **intégré dans `CreationBoites`**, destiné à mettre en forme des **exercices ou activités** dans un polycopié, avec :

- un **numéro** automatique (compteur `cbhlb` par défaut, ou un compteur dédié si on passe par `\CreationExoStyleH`);
- un **niveau de difficulté** affiché par des pastilles (●●○○) et une couleur dédiée;
- un **type** optionnel (cours, QCM, vrai/faux, Python...) affiché en sous-titre, avec icône, **librement extensible** par l'utilisateur;
- une **couleur de sous-titre** elle aussi extensible, avec un mode couleur auto qui peut aligner automatiquement chaque type sur la couleur du niveau;
- un simple **filet horizontal** coloré sous le titre (pas de cadre) pour rester léger visuellement.

## Remarque

Le nom `cbhlb` est une *invention* (« `CreationBoites High Level Box` », ou simplement un héritage de développement...), c'est le préfixe utilisé pour toutes les macros et couleurs internes du système, afin d'éviter toute collision avec d'autres packages, y compris les autres modules de `CreationBoites`.

## 2 Analyse et explication du code source

### 2.1 Dépendances

Le module est intégré directement dans `CreationBoites.sty` : aucune dépendance supplémentaire à charger soi-même, le `.sty` embarque déjà tout ce qu'il faut :

#### Code

```
\RequirePackage{simplekv}           % gestion des trousseaux de clés
\RequirePackage{tcolorbox}
\tcbuselibrary{breakable,skins}
\RequirePackage{luclide-icons}     % icônes (sous-titres)
\RequirePackage{pifont}           % pastilles de niveau (\ding)
```

## Remarque

`pifont` a remplacé `wasysym` dans cette version. Les deux donnaient des pastilles homogènes, mais `wasysym` a tendance à entrer en collision avec `amsmath` (la commande `\iint` est définie par les deux packages). `pifont` n'a aucune commande mathématique de ce genre, donc **zéro risque de conflit**.

### 2.2 Variables et trousseau de paramètres globaux

On peut commencer par définir des variables utiles (compteur, hauteur, etc)

#### Code

```
\def\cbhlbstrut{(qÉ/B)}%pour avoir toujours une hauteur homogène
\ifundefined{c@cbhlb}{\newcounter{cbhlb}}{}%
\setcounter{cbhlb}{0}
\newlength\lhbbrtwidthrule
```

On passe par `simplekv` pour définir le trousseau, `paramcbh1b` :

#### Code

```
\defKV[paramcbh1b]{%
  eptrait      = \setlength\lhbwidthrule{#1},%
  police num   = \def\cbh1bpolicetitre{#1},%
  police sstitre = \def\cbh1bpolicesoustitre{#1},%
  police niveau = \def\cbh1bpoliceniveau{#1},%
  label exo    = \def\cbh1blabelexo{#1},%
  apres num    = \def\cbh1bsuitelabelexo{#1},%
  compteur     = \def\cbh1bcount{#1}%
}
\setKVdefault[paramcbh1b]{%
  eptrait      = 0.375mm,%
  police num   = \Large\bfseries\sffamily,%
  police sstitre = \footnotesize\bfseries\sffamily,%
  police niveau = \scriptsize\bfseries\sffamily,%
  label exo    = {},%
  apres num    = {},%
  compteur     = cbh1b,%
  couleurs     = true,%
  couleur auto = true
}
```

#### À retenir

`\@ifundefined{c@cbh1b}{\newcounter{cbh1b}}{ }` protège la création du compteur : si `CreationBoites` est chargé deux fois, ou si un autre module a déjà créé `cbh1b`, on ne replante pas dessus.

Ce trousseau `paramcbh1b` centralise les réglages **globaux** (valables pour *tous* les exercices suivants, jusqu'au prochain réglage).

| Clé                         | Effet                                                   | Défaut                                       |
|-----------------------------|---------------------------------------------------------|----------------------------------------------|
| <code>eptrait</code>        | épaisseur du filet sous le titre                        | 0.375mm                                      |
| <code>police num</code>     | police du numéro d'exercice                             | <code>\Large\bfseries\sffamily</code>        |
| <code>apres num</code>      | complément après le numéro                              | <code>{}</code> (vide)                       |
| <code>police sstitre</code> | police du sous-titre/type                               | <code>\footnotesize\bfseries\sffamily</code> |
| <code>police niveau</code>  | police du libellé + pastilles de niveau                 | <code>\scriptsize\bfseries\sffamily</code>   |
| <code>label exo</code>      | texte avant le numéro (ex. Exercice )                   | <code>{}</code> (vide)                       |
| <code>compteur</code>       | nom du compteur L <sup>A</sup> T <sub>E</sub> X utilisé | <code>cbh1b</code>                           |
| <code>couleurs</code>       | booléen — active/désactive toutes les couleurs          | <code>true</code>                            |
| <code>couleur auto</code>   | booléen — voir §2.6                                     | <code>true</code>                            |

## 2.3 Couleurs

#### Code

```
%--les couleurs de niveau
\definecolor{cbh1bcoulcours}{HTML}{6fb8cb}
\definecolor{cbh1bcoulfacile}{HTML}{61ab68}
\definecolor{cbh1bcoulmoyen}{HTML}{df9f3d}
\definecolor{cbh1bcouldifficile}{HTML}{cc5745}
\definecolor{cbh1bcouleexpert}{HTML}{824aa7}
\colorlet{cbh1bsoustitre}{gray}
%--les couleurs de sous-titres
\definecolor{cbh1bcoulqcm}{HTML}{bb5c25}
\definecolor{cbh1bcoulvraifaux}{HTML}{bb5c25}
\definecolor{cbh1bcoulpython}{HTML}{507fb4}
\definecolor{cbh1bcoulimportant}{HTML}{bf0000}
\definecolor{cbh1bcoultableur}{HTML}{61ab68}
\definecolor{cbh1bcoulgeogebra}{HTML}{bf2f64}
```

Chaque niveau de difficulté (cours, facile, moyen, difficile, expert) a sa propre couleur définie par `cbhlbcoul<niveau>`, utilisée à la fois pour le numéro et pour le filet horizontal. Les couleurs `cbhlbcoulqcmvraifaux`, `cbhlbcoulpython`, `cbhlbcoulimportant` servent aux **types** de sous-titre prédéfinis : toutes suivent la même convention de nommage `cbhlbcoul<type>`.

#### Remarque

Cette convention `cbhlbcoul<XXX>` systématique (couleurs *et* sous-titres) est ce qui rend tout le système extensible sans toucher au `.sty` (voir §4.5).

## 2.4 Pastilles de niveau (pifont)

#### Code

```
\def\cbhlbnivA{\raisebox{\depth}{\ding{109}\ding{109}\ding{109}\ding{109}}}
\def\cbhlbnivB{\raisebox{\depth}{\ding{108}\ding{109}\ding{109}\ding{109}}}
\def\cbhlbnivC{\raisebox{\depth}{\ding{108}\ding{108}\ding{109}\ding{109}}}
\def\cbhlbnivD{\raisebox{\depth}{\ding{108}\ding{108}\ding{108}\ding{109}}}
\def\cbhlbnivE{\raisebox{\depth}{\ding{108}\ding{108}\ding{108}\ding{108}}}

\def\cbhlblabelnivA{}
\def\cbhlblabelnivB{FACILE}
\def\cbhlblabelnivC{MOYEN}
\def\cbhlblabelnivD{DIFFICILE}
\def\cbhlblabelnivE{EXPERT}
```

#### Remarque

`\ding{108}` (plein) et `\ding{109}` (vide) viennent de la fonte Zapf Dingbats (pifont), pas des symboles mathématiques  $\bullet$  ou  $\circ$  : ces derniers dépendent de la police maths active et ne sont pas toujours calibrés de façon homogène entre eux. Issus de la même fonte dédiée, `\ding{108}` et `\ding{109}` ont un rendu garanti homogène, quelle que soit la police de texte ou de maths choisie par ailleurs. `\raisebox{\depth}{...}` recale verticalement les pastilles pour qu'elles s'alignent proprement sur la ligne de base du texte environnant.

## 2.5 Style de boîte `tcolorbox`

#### Code

```
\tcbset{basecbhlb/.style={
  enhanced,empty,%
  breakable,%
  size=small,boxsep=0mm,boxrule=0mm,left=0mm,right=0mm,
  before skip=0.75\baselineskip,after skip=0.75\baselineskip
}%
}
```

Le style `basecbhlb` est volontairement **minimaliste** : `empty` et `boxrule=0mm` suppriment tout cadre — seul le **filet horizontal** généré manuellement (via `\hrule`) marque visuellement le début de l'exercice. `breakable` autorise la boîte à se couper sur plusieurs pages.

## 2.6 La couleur de sous-titre extensible : \cbhlbtextcoul

### Code

```
\newcommand\cbhlbtextcoul[2]{%
  \ifboolKV[paramcbhlb]{couleur auto}%
    {\textcolor{cbhlbcoulcurrent}{#2}}%
  {%
    \@ifundefinedcolor{cbhlbcoul#1}%
      {\textcolor{cbhlbcoulcurrent}{#2}}%
      {\textcolor{cbhlbcoul#1}{#2}}%
    }%
}
```

C'est le cœur du système de couleurs extensible. `\cbhlbtextcoul{#1}{#2}` affiche le contenu #2 (typiquement un sous-titre) dans une couleur qui dépend du type #1 :

- si couleur auto=true (**réglage par défaut du package**) : **tout** sous-titre, prédéfini ou non, s'affiche dans la couleur **du niveau** de l'exercice (cbhlbcoulcurrent) — aucune exception ;
- si couleur auto=false : on regarde si une couleur cbhlbcoul<type> existe (via la primitive native \@ifundefinedcolor) :
  - **si elle existe** → on l'utilise (couleur **fixe**, indépendante du niveau) ;
  - **si elle n'existe pas** → on retombe quand même sur la couleur du niveau.

### À retenir

`\@ifundefinedcolor` est une primitive **native** de xcolor : elle teste proprement l'existence d'une couleur sans dépendre de l'implémentation interne du package (contrairement à un test manuel sur color@<nom>).

### Remarque

La logique auto est testée **avant** l'existence de la couleur : avec couleur auto=true, une couleur cbhlbcoul<type> définie par l'utilisateur sera donc **ignorée**. C'est un choix de conception assumé : couleur auto est un interrupteur global, sans exception, pour qu'une couleur dédiée prenne le dessus, il faut passer en couleur auto=false.

## 2.7 Sous-titres prédéfinis (type d'exercice)

### Code

```
\def\cbhlbssttitreours{%
  {\lucideicon[height=auto]{heart}\,COURS}%
}
\def\cbhlbssttitreqcm{%
  {\lucideicon[height=auto]{scale}\,QCM}%
}
\def\cbhlbssttitreveraifaux{%
  {\lucideicon[height=auto]{square-check-big}\,VRAI/FAUX}%
}
\def\cbhlbssttitrepython{%
  {\lucideicon[height=auto]{code-xml}\,PYTHON}%
}
\def\cbhlbssttitreimportant{%
  {\lucideicon[height=auto]{biceps-flexed}\,IMPORTANT}%
}
\def\cbhlbssttitretableur{%
  {\lucideicon[height=auto]{sheet}\,TABLEUR}%
}
\def\cbhlbssttitregeogebra{%
  {\lucideicon[height=auto]{tangent}\,GEOGEBRA}%
}
```

Chaque type prédéfini suit la convention de nommage `\cbhlbsstitle<nom>` (en minuscules, sans espace). C'est cette convention que `\cbhlbgensoustitle` va exploiter pour savoir si un type donné a une présentation enrichie (icône) ou pas.

#### Remarque

Contrairement à une version antérieure, ces macros ne contiennent **plus aucun** `\textcolor` en dur : la couleur est désormais entièrement déléguée à `\cbhlbtextcoul`, appelée depuis `\cbhlbgensoustitle`.

C'est ce découplage qui permet à couleur auto de s'appliquer uniformément, y compris à des types personnalisés.

## 2.8 Génération du sous-titre

#### Code

```
\newcommand\cbhlbgensoustitle[1]{%
  \intcbtestsubstr{#1}{\$}%
  {%
    \intcbafterchar{#1}{\$}[\cbhlbsstmp]%
    {~~\cbhlbpoliciesoustitle\vphantom{\cbhlbstrut}\cbhlbsstmp}%
  }%
  {%
    \@ifundefined{cbhlbsstitle#1}{%
      {~~\cbhlbpoliciesoustitle\vphantom{\cbhlbstrut}\cbhlbtextcoul{#1}{#1}}}%
      {~~\cbhlbpoliciesoustitle\vphantom{\cbhlbstrut}%
        \cbhlbtextcoul{#1}{\csname cbhlbsstitle#1\endcsname}}}%
  }%
}
```

C'est le cœur de l'extensibilité du système :

- si l'argument commence par \$, on passe ce qui suit en lecture/affichage direct (texte ou code  $\LaTeX$  libre, sans passer par `\cbhlbtextcoul`);
- sinon, `\@ifundefined{cbhlbsstitle#1}` teste si la macro `\cbhlbsstitle<type>` existe;
  - **si elle existe** (type prédéfini, ex. python) → on l'utilise, en lui appliquant la macro de coloration `\cbhlbtextcoul{#1}{...}` (icône + couleur);
  - **si elle n'existe pas** (type libre, ex. IMPORTANT mal orthographié, ou un type dont seule la couleur a été définie) → le texte **brut** #1 est affiché, mais **toujours** passé par la macro `\cbhlbtextcoul{#1}{#1}` : il bénéficie donc de la couleur dédiée `cbhlbcoul<type>`, si elle existe.

#### À retenir

Cette double bascule (icône *et* couleur, chacune indépendamment optionnelle) permet d'ajouter un nouveau type **progressivement** : une couleur seule, une icône seule, ou les deux, et ce sans jamais toucher au code de CreationBoites (cf. §4.5).

## 2.9 Construction du titre

#### Code

```
\newcommand\cbhlbinttitre[4]{%
  {\cbhlbpolicetitre\vphantom{\cbhlbstrut}%
    \textcolor{#1}{\cbhlblabelexo\arabic{\cbhlbcount}\cbhlbsuitelabelexo}}%lbl+num
  {\cbhlbgensoustitle{#4}}%sous-titre/thème
  \hfill{}%
  {\cbhlbpoliceniveau\vphantom{\cbhlbstrut}%
    \textcolor{cbhlbsoustitle}{#2}\enspace\textcolor{#1}{#3}}%niveau
}
```

`\cbhlbinttitre` assemble visuellement le titre en quatre blocs :

| Argument | Rôle                                                   |
|----------|--------------------------------------------------------|
| #1       | couleur du niveau (ex. <code>cbhlbcoulfacile</code> )  |
| #2       | libellé textuel du niveau (ex. FACILE)                 |
| #3       | pastilles de difficulté (ex. <code>\cbhlbnivB</code> ) |
| #4       | type, transmis à <code>\cbhlbgensoustitre</code>       |

Le numéro et le type à gauche, le niveau textuel et ses pastilles à droite (séparés par un `\hfill`). Le `\vphantom{\cbhlbstrut}` (présent dans chaque bloc) garantit une hauteur de ligne homogène, quel que soit le contenu réellement affiché.

## 2.10 Sélection du niveau (1 à 4, ou 0 pour le cours)

### Code

```
\newcommand\cbhlbgentitre[2][1]{%
  %#1 = niveau
  %#2 = sous-titre
  \stepcounter{\cbhlbcount}%
  \ifnum#1=0\relax\cbhlbinttitre{cbhlbcoulcours}{\cbhlblabelnivA}{\cbhlbnivA}{#2}\fi%
  \ifnum#1=1\relax\cbhlbinttitre{cbhlbcoulfacile}{\cbhlblabelnivB}{\cbhlbnivB}{#2}\fi%
  \ifnum#1=2\relax\cbhlbinttitre{cbhlbcoulmoyen}{\cbhlblabelnivC}{\cbhlbnivC}{#2}\fi%
  \ifnum#1=3\relax\cbhlbinttitre{cbhlbcouldifficile}{\cbhlblabelnivD}{\cbhlbnivD}{#2}\fi%
  \ifnum#1=4\relax\cbhlbinttitre{cbhlbcoulexpert}{\cbhlblabelnivE}{\cbhlbnivE}{#2}\fi%
  %
  \ifnum#1=0\relax\textcolor{cbhlbcoulcours}{\hrule height \lhbrtwidththrule}\fi%
  \ifnum#1=1\relax\textcolor{cbhlbcoulfacile}{\hrule height \lhbrtwidththrule}\fi%
  \ifnum#1=2\relax\textcolor{cbhlbcoulmoyen}{\hrule height \lhbrtwidththrule}\fi%
  \ifnum#1=3\relax\textcolor{cbhlbcouldifficile}{\hrule height \lhbrtwidththrule}\fi%
  \ifnum#1=4\relax\textcolor{cbhlbcoulexpert}{\hrule height \lhbrtwidththrule}\fi%
  %
  \medskip
}
```

`\cbhlbgentitre` incrémente le compteur *une seule fois* (`\stepcounter`), puis choisit le bon jeu couleur/libellé/pastilles selon le **niveau** (#1, de 0 à 4), et trace enfin le filet horizontal dans la couleur correspondante.

| Niveau | Couleur            | Libellé   | Pastilles                               |
|--------|--------------------|-----------|-----------------------------------------|
| 0      | cbhlbcoulcours     | (aucun)   | <code>\cbhlbnivA</code> ( <i>mmmm</i> ) |
| 1      | cbhlbcoulfacile    | FACILE    | <code>\cbhlbnivB</code> ( <i>lmmm</i> ) |
| 2      | cbhlbcoulmoyen     | MOYEN     | <code>\cbhlbnivC</code> ( <i>llmm</i> ) |
| 3      | cbhlbcouldifficile | DIFFICILE | <code>\cbhlbnivD</code> ( <i>lllm</i> ) |
| 4      | cbhlbcoulexpert    | EXPERT    | <code>\cbhlbnivE</code> ( <i>llll</i> ) |

## 3 L'environnement exocbhlb

### 3.1 La macro de création de l'environnement

#### Code

```
\defKV[basecbhlb]{niveau=\def\cbhlbniveau{#1},type=\def\cbhlbtype{#1}}
\setKVdefault[basecbhlb]{niveau=1,type={}}

\NewDocumentEnvironment{exocbhlb}{0{} D<>{} }%
{%
  \restoreKV[basecbhlb]%
  \setKV[basecbhlb]{#1}%
  \ifboolKV[paramcbhlb]{couleurs}%
  {%
    {%
      \colorlet{cbhlbcoulcours}{black}%
      \colorlet{cbhlbcoulfacile}{black}%
      \colorlet{cbhlbcoulmoyen}{black}%
      \colorlet{cbhlbcouldifficile}{black}%
      \colorlet{cbhlbcouleexpert}{black}%
      \colorlet{cbhlbsoustitre}{darkgray!85}%
      \colorlet{cbhlbcoulqcm}{darkgray!85}%
      \colorlet{cbhlbcoulvraifaux}{darkgray!85}%
      \colorlet{cbhlbcoulpython}{darkgray!85}%
      \colorlet{cbhlbcoulimportant}{darkgray!85}%
      \colorlet{cbhlbcoultableur}{darkgray!85}%
      \colorlet{cbhlbcoultableur}{cbhlbcoulgeogebra!85}%
    }%
    \ifnum\cbhlbniveau=0\relax\colorlet{cbhlbcoulcurrent}{cbhlbcoulcours}\fi%
    \ifnum\cbhlbniveau=1\relax\colorlet{cbhlbcoulcurrent}{cbhlbcoulfacile}\fi%
    \ifnum\cbhlbniveau=2\relax\colorlet{cbhlbcoulcurrent}{cbhlbcoulmoyen}\fi%
    \ifnum\cbhlbniveau=3\relax\colorlet{cbhlbcoulcurrent}{cbhlbcouldifficile}\fi%
    \ifnum\cbhlbniveau=4\relax\colorlet{cbhlbcoulcurrent}{cbhlbcouleexpert}\fi%
    \tcbsset{reset}%
    \tcolorbox[basecbhlb,#2]%
    \cbhlbgentitre[\cbhlbniveau]{\cbhlbtype}%
  }%
  {%
    \endtcbox%
  }%
}
```

Deux jeux d'arguments, comme pour les autres boîtes `CreationBoites` :

- `[niveau=...,type=...]` (entre crochets) : les deux clés *propres* à `exocbhlb` (niveau 1 et type vide par défaut);
- `<options tcbbox>` (entre chevrons) : n'importe quelle option `tcolorbox` supplémentaire, transmise directement à `\tcolorbox[basecbhlb,#2]`.

`\colorlet{cbhlbcoulcurrent}{...}` capture la couleur du niveau **courant** dans une unique couleur de référence — c'est elle que `\cbhlbtextcoul` va utiliser en mode couleur `auto=true` ou en fallback.

### 3.2 Un compteur dédié : `\CreationExoStyleH`

#### Code

```
\NewDocumentCommand\CreationExoStyleH{ m m }{%
  %1 = nom environnement // %2 = compteur (avec test)
  \@ifundefined{c@#2}{\newcounter{#2}}{}%
  \setcounter{#2}{0}%
  \NewDocumentEnvironment{#1}{0{} D<>{} }%
  {%
    \def\cbhlbcount{#2}%           compteur figé par substitution
    \restoreKV[basecbhlb]%
    %...                           (même corps que exocbhlb)
  }
```



`\CreationExoStyleH` génère un **nouvel environnement**, doté de son **propre compteur indépendant** (créé et initialisé à 0 si besoin). Le compteur est **figé par substitution** de #2 dans `\def\cbhlbcount{#2}` : chaque environnement généré porte ainsi son propre compteur en dur dans son code, sans jamais passer par le trousseau global `paramcbhlb` (qui resterait, lui, partagé entre tous les styles).

#### Remarque

Usage typique : séparer la numérotation des exercices de TD de celle des exercices d’approfondissement, sans qu’ils interfèrent.

```
\CreationExoStyleH{exoTD}{cptTD}
\CreationExoStyleH{exoAppro}{cptAppro}
```

```
\begin{exoTD}[niveau=1] ... \end{exoTD} % compteur cptTD
\begin{exoAppro}[niveau=4] ... \end{exoAppro} % compteur cptAppro, indépendant
```

### 3.3 Mise à jour des paramètres globaux

#### Code

```
\newcommand\paramsexocbhlb[1]{%
  \ifthenelse{\equal{#1}{}}{%
    {%valeurs par défaut
      \setKVdefault[paramcbhlb]{%
        eptrait      = 0.375mm,%
        police num   = \Large\bfseries\sffamily,%
        police ssttre = \footnotesize\bfseries\sffamily,%
        police niveau = \scriptsize\bfseries\sffamily,%
        label exo    = {},%
        apres num    = {},%
        compteur     = cbhlb,%
        couleurs     = true,%
        couleur auto  = false
      }
    }%
    {%nouvelles valeurs par défaut
      \setKVdefault[paramcbhlb]{#1}%
    }%
    \restoreKV[paramcbhlb]%
  }
}
```

`\paramsexocbhlb` applique les nouvelles clés par défaut, *puis* les **restaure immédiatement** via la macro de remise à zéro `\restoreKV[paramcbhlb]`.

#### À retenir

Ce `\restoreKV[paramcbhlb]` final est **indispensable** : `\setKVdefault` seul ne fait que changer la valeur *par défaut* d’une clé pour un futur `\restoreKV`, sans jamais l’appliquer elle-même. Sans cette ligne, un changement comme `couleur auto=true` ne produirait **aucun effet visible**, peu importe combien de fois on l’appelle.

Avec un argument vide (`\paramsexocbhlb{}`), on revient aux réglages par défaut **du package** (notez que `couleur auto` y vaut `false`, alors que la valeur déclarée au chargement du `.sty` était `true`, donc un appel à `\paramsexocbhlb{}` n’est donc pas un « reset complet », juste un retour aux réglages « habituels » d’utilisation).

## 4 Utilisation

### 4.1 Paramétrage global

#### Code

```
\paramsexocbhlb{%  
  eptrait      = 0.375mm,%  
  police num   = \Large\bfseries\sffamily,%  
  police ssttre = \footnotesize\bfseries\sffamily,%  
  police niveau = \scriptsize\bfseries\sffamily%  
}
```

### 4.2 Syntaxe de base

#### Code

```
\begin{exocbhlb}[niveau=...,type=...]  
%contenu de l'exercice  
\end{exocbhlb}
```

#### 4.2.1 Les cinq niveaux, sans type

#### Code

```
\begin{exocbhlb}[niveau=0/1/2/3/4]  
\lipsum[1][1-2]  
\end{exocbhlb}
```

#### Sortie

1

○○○○

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

2

FACILE ●○○○

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

3

MOYEN ●●○○

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

4

DIFFICILE ●●●○

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

5

EXPERT ●●●●

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

## 4.2.2 Types prédéfinis (avec icône)

### Code

```
\begin{exocbhlb}[niveau=2,type=qcm]
\lipsum[2][1-2]
\end{exocbhlb}

\begin{exocbhlb}[niveau=3,type=vraifaux]
\lipsum[2][1-2]
\end{exocbhlb}

\begin{exocbhlb}[niveau=4,type=python]
\lipsum[2][1-2]
\end{exocbhlb}

\begin{exocbhlb}[niveau=4,type=important]
\lipsum[2][1-2]
\end{exocbhlb}

\begin{exocbhlb}[niveau=1,type=tableur]
\lipsum[2][1-2]
\end{exocbhlb}

\begin{exocbhlb}[niveau=2,type=geogebra]
\lipsum[2][1-2]
\end{exocbhlb}
```

### Sortie

#### 6 QCM

MOYEN ●●○○

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

#### 7 VRAI/FAUX

DIFFICILE ●●●○

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

#### 8 PYTHON

EXPERT ●●●●

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

#### 9 IMPORTANT

EXPERT ●●●●

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

#### 10 TABLEUR

FACILE ●○○○

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

#### 11 GEOGEBRA

MOYEN ●●○○

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

### Remarque

Par défaut (couleur `auto=true`), QCM/VRAI/FAUX/PYTHON s'affichent dans la couleur **du niveau** de l'exercice, et non dans leur couleur dédiée (`cbhlbcoulqcm`, `cbhlbcoulpython`) (voir §4.4) pour les faire ressortir avec leur propre teinte.

### 4.2.3 Type libre (texte brut ou texte formaté)

#### Code

```
\begin{exocbhlb}[niveau=1,type=ATTENTION]
\lipsum[3][1-2]
\end{exocbhlb}
```

#### Sortie

### 12 ATTENTION

FACILE ●○○○

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.

#### Code

```
\begin{exocbhlb}[niveau=1,type=${\textcolor{red}{ATTENTION}}]
\lipsum[3][1-2]
\end{exocbhlb}
```

#### Sortie

### 13 ATTENTION

FACILE ●○○○

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.

#### À retenir

Le préfixe `type=$...` bascule en lecture directe : le contenu qui suit `$` est affiché **tel quel**, en  $\text{\LaTeX}$  libre, sans passer par `\cbhlbtextcoul`. C'est la solution la plus rapide pour un besoin **ponctuel**, qui ne mérite pas de créer une macro `\cbhlbsstitle<nom>` dédiée.

## 4.3 Désactiver toutes les couleurs

#### Code

```
\paramsexocbhlb{%
  label exo={Exercice\enspace},%
  apres num={\,$\cdot$},%
  couleurs=false%
}
\begin{exocbhlb}[niveau=2]<width=0.5\linewidth>
\lipsum[1][3-4]
\end{exocbhlb}
```

#### Sortie

### Exercice 14.

MOYEN ●●○○

Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

## 4.4 Couleurs dédiées par type : couleur auto=false

Pour que les types prédéfinis (qcm, vraifaux, python, important) retrouvent leur **couleur propre**, indépendamment du niveau :

#### Code

```
\paramsexocbhlb{couleur auto=false}  
  
\begin{exocbhlb}[niveau=1,type=python]  
\lipsum[1][1]  
\end{exocbhlb}  
  
\begin{exocbhlb}[niveau=4,type=tableur]  
\lipsum[1][1]  
\end{exocbhlb}
```

#### Sortie

**15** </ PYTHON

FACILE ●○○○

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

**16** </ PYTHON

EXPERT ●●●●

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

#### Remarque

PYTHON reste maintenant en bleu (cbhlbcoulpython) dans les deux boîtes, **malgré** des niveaux différents (numéro et filet, eux, suivent toujours le niveau).

## 4.5 Ajouter un nouveau type

Quatre façons de procéder, du plus simple au plus complet, en sachant qu'**aucune ne nécessite de toucher au .sty**.

### 4.5.1 Juste une couleur dédiée (sans icône)

#### Code

```
\colorlet{cbhlbcouloral}{violet}  
\paramsexocbhlb{couleur auto=false}  
  
\begin{exocbhlb}[niveau=3,type=oral]  
\lipsum[1][1]  
\end{exocbhlb}
```

#### Sortie

**17** oral

DIFFICILE ●●●○

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Le texte oral est affiché brut (aucune macro \cbhlbsstitleoral définie), mais **coloré en violet** — la couleur dédiée est bien retrouvée par \cbhlbtextcoul, même sans icône associée.

### 4.5.2 Juste une icône (suit la couleur du niveau)

#### Code

```
\def\cbhlbssttitregeom{\{\lucideicon[height=auto]{calculator}\,GEOM}}
\paramsexocbhlb{couleur auto=true}

\begin{exocbhlb}[niveau=2,type=geom]
\lipsum[1][1]
\end{exocbhlb}
```

#### Sortie

18  GEOM

MOYEN ●●○○

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### 4.5.3 Icône + couleur dédiée

#### Code

```
\def\cbhlbssttitreoral{\{\lucideicon[height=auto]{mic}\,ORAL}}
\colorlet{cbhlbcouloral}{teal}
\paramsexocbhlb{couleur auto=false}

\begin{exocbhlb}[niveau=4,type=oral]
\lipsum[1][1]
\end{exocbhlb}
```

#### Sortie

19  ORAL

EXPERT ●●●●

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### 4.5.4 Macro de génération automatique

La macro suivante permet de créer *automatiquement* la couleur et la macro pour un sous-titre personnalisé, en précisant le suffixe, la couleur et la macro :

#### Code

```
\NewDocumentCommand\gencbhlbsoustitre{ m m m }{%
% #1 = suffixe (ex: oral)
% #2 = couleur (ex: violet)
% #3 = contenu (ex: \lucideicon[height=auto]{mic}\,ORAL)
\colorlet{cbhlbcoul#1}{#2}%
\expandafter\def\csname cbhlbssttitre#1\endcsname{#3}%
}
```

#### Code

```
\gencbhlbsoustitre{ggb}%
{cyan}%
{\lucideicon[height=auto]{calculator}\,GGB}
```

#### Sortie

20  GGB

FACILE ●○○○

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

#### 4.5.5 Consignes d'utilisation

##### À retenir

Le nom passé à `type=...` doit correspondre **exactement** (casse comprise) à la fin du nom de macro `\cbhlbsstitre<nom>` *et* de couleur `cbhlbcoul<nom>`. Par exemple `type=Oral` (majuscule) ne déclenchera **ni** `\cbhlbsstitreoral` **ni** `cbhlbcouloral` (minuscules) : le texte `Oral` brut serait affiché, dans la couleur du niveau.

| Vous définissez                             | Icône | Couleur dédiée | Résultat                                                         |
|---------------------------------------------|-------|----------------|------------------------------------------------------------------|
| rien                                        | ✗     | ✗              | texte brut, couleur du niveau                                    |
| <code>cbhlbcoul&lt;type&gt;</code> seule    | ✗     | ✓              | texte brut, couleur dédiée ( <i>si <code>auto=false</code></i> ) |
| <code>cbhlbsstitre&lt;type&gt;</code> seule | ✓     | ✗              | icône, couleur du niveau                                         |
| les deux                                    | ✓     | ✓              | icône, couleur dédiée ( <i>si <code>auto=false</code></i> )      |

## 5 Récapitulatif

| Commande / Environnement                          | Rôle                                                                                    |
|---------------------------------------------------|-----------------------------------------------------------------------------------------|
| <code>\paramsexocbhlb{clés}</code>                | réglages globaux (filet, polices, label, compteur, couleurs)                            |
| <code>exocbhlb[niveau=,type=]&lt;tcbox&gt;</code> | boîte d'exercice (compteur partagé cbhlb)                                               |
| <code>\CreationExoStyleH{nom}{compteur}</code>    | nouvel environnement avec compteur dédié                                                |
| <code>\cbhlbsstitle&lt;nom&gt;</code>             | icône d'un nouveau type                                                                 |
| <code>cbhlbcoul&lt;nom&gt;</code>                 | couleur dédiée d'un nouveau type ( <code>\colorlet</code> )                             |
| <code>couleur auto</code>                         | <code>true</code> : tout suit le niveau / <code>false</code> : couleurs dédiées actives |

- `simplekv` : <https://ctan.org/pkg/simplekv>
- `tcolorbox` : <https://ctan.org/pkg/tcolorbox>
- `lucide-icons` : <https://ctan.org/pkg/lucide-icons>
- `pifont` : <https://ctan.org/pkg/pifont>