L'extension rdcrep

Produire des boîtes correctives.

Ce package permet de composer des cadres de texte (boîtes) destinés à recevoir les réponses des élèves. Le document final peut être généré en activant le mode *correction* ce qui entraîne les remplissage automatique de l'ensemble des boîtes par le texte correctif pré-établi.

La couleur suivante est prédéfinie par l'extension :

seyescol

Chargement de l'extension

De la manière la plus classique qui soit, on charge l'extension à l'aide de la commande :

```
\usepackage{rdcrep}
```

Aucune option de chargement n'est proposée.

Cette extension est dépendante des packages **calc.sty**, **tikz.sty**, **tcolorbox.sty**, **listofitems.sty**, **pifont.sty** et **nicematrix.sty**.

Plusieurs compilations successives seront souvent nécessaires avant d'obtenir la mise en forme souhaitée.

Modification des valeurs par défaut des clés

\setrdcrep{⟨*paramètres*⟩}

Cette commande permet de modifier les valeurs attribuées par défaut aux différentes clés de l'extension.
Il est également permis, à l'aide de cette commande, de créer des styles utilisateur pouvant être réinvestis à volonté.

```
\setrdcrep{correction=false}
$2x+\tccrep{1cm}{3x}=5x$

\setrdcrep{correction=true}
$2x+\tccrep{1cm}{3x}=5x$
```

$2x + \quad = 5x$

$2x + 3x = 5x$

Les quatre principaux types de boîtes

Ce paragraphe présente, de manière rapide, les trois principaux types de boîtes que l'extension permet de créer. Des informations complémentaires quant à leur fonctionnement et à la façon de modifier les réglages par défaut seront fournies dans les paragraphes suivants.

```
\begin{crep}[⟨options⟩]
  ⟨contenu d'environnement⟩
\end{crep}
```

Cet environnement crée une boîte du type `tcolorbox` sensible au mode correctif (autrement dit, cet environnement produit des résultats différents selon la valeur de la clé `/crep/correction`). Les ⟨*options*⟩ permises sont les clés habituelles de la famille `/tcb/` ainsi que certaines nouvelles clés proposées par l'extension.

Par défaut, les boîtes du type `crep` se voient affecter le style (modifiable par l'utilisateur) défini ci-contre.

```
\tcbset{%
    crep default style/.style={
            enhanced jigsaw,
            before skip balanced=0.25\baselineskip plus 2pt,
            after skip balanced=0.5\baselineskip plus 2pt,
            notitle,
            valign=top, halign=justify,
            colback=gray!10, colframe=black,
            boxsep=7pt, top=0pt, bottom=0pt, left=0pt, right=0pt,
            middle=0pt, sidebyside gap=0pt, lower separated=false
    }
}
```

```
\setrdcrep{correction=false}
Quelle est la définition de \frquote{hauteur} dans un
     triangle ?
\begin{crep}
Dans un triangle, une hauteur est une droite qui
    passe par un sommet et qui est perpendiculaire au
     côté opposé à ce sommet.
\end{crep}
```

Quelle est la définition de « hauteur » dans un triangle ?

`\tccrep[`⟨*options*⟩`]{`⟨*longueur*⟩`}{`⟨*contenu*⟩`}`

Cette commande permet de produire des boîtes du type `tcbox` sensibles au mode correctif. La largeur de la boîte est donnée par le paramètre ⟨*longueur*⟩. Par défaut, le contenu est centré dans la boîte.

Les boîtes produites par la commande `\tccrep` se voient affecter, par défaut, le style (modifiable par l'utilisateur) défini ci-contre.

```
\tcbset{%
    tccrep default style/.style={
            on line,
            enhanced jigsaw,
            colback=gray!10,
            boxsep=5pt,
            top=0pt, bottom=0pt, left=0pt, right=0pt,
            sharp corners,
            frame engine=empty,
            boxrule=0pt
    }
}
```

```
\setrdcrep{correction=true}
Si un point est \tccrep{4cm}{équidistant}
des extrémités d'un segment alors ce point
appartient à la
\setrdcrep{correction=false}
\tccrep{4cm}{médiatrice} de ce segment.
```

Si un point est **équidistant** des extrémités d'un segment alors ce point appartient à la de ce segment.

`\tcautocrep[`⟨*options*⟩`]{`⟨*contenu*⟩`}`

La commande `\tcautocrep` produit des boîtes identiques à celles générées par la commande `\tccrep` mais la largeur de la boîte n'a pas à être précisée par l'utilisateur, celle-ci étant automatiquement calculée par l'extension. Par défaut, les boîtes ont une largeur égale au double de la largeur naturelle de leur contenu. Ce réglage peut être modifié à l'aide de l'une des clés `/crep/auto factor` ou `/tcb/crep/auto factor`.

```
\setrdcrep{correction=true}
Si un point est \tcautocrep{équidistant}
des extrémités d'un segment alors ce point
appartient à la \tcautocrep{médiatrice} de
ce segment.
```

Si un point est **équidistant** des extrémités d'un segment alors ce point appartient à la **médiatrice** de ce segment.

`\tcfillcrep`[⟨*options*⟩]{⟨*contenu*⟩}

Cette commande permet de produire des boîtes du type `tcbox` sensibles au mode correctif. La largeur de la boîte s'adapte automatiquement à la place disponible sur la ligne (à la manière de la commande `\dotfill`). Par défaut, le contenu dans la boîte est aligné à gauche.
Pour obtenir le résultat souhaité avec cette commande, au moins deux compilations successives sont nécessaires.
La commande `\tcfillcrep` partage son style défini par défaut avec la commande `\tccrep`.

```
Nom : \tcfillcrep{}

Prénom : \tcfillcrep{} Classe : \tccrep{1cm}{}
```

Nom :

Prénom : Classe :

Remarque : lorsque les commandes `\tccrep` et `\tcfillcrep` sont utilisées en mode mathématiques, il ne faut pas placer le contenu de ces boîtes entre dollars (ces commandes détectent le mode dans lequel elles sont utilisées et formattent le contenu en conséquence).

```
\setrdcrep{correction}
Exemple :

\[ (a+b)^2 = \tccrep{3cm}{a^2 + 2ab + b^2} \]

Ou encore : $(2x+3)^2=\tcfillcrep{4x^2+12x+9}$
```

Exemple :

$$(a+b)^2 = a^2+2ab+b^2$$

Ou encore : $(2x+3)^2 = 4x^2+12x+9$

Les clés globales

Les clés décrites dans ce paragraphe appartiennent à la famille `/crep/` et sont destinées à être utilisées avec la commande `\setrdcrep`. Les modifications des valeurs des clés effectuées avec cette commande sont globales ou limitées au groupe au sein duquel la commande est appelée.

`/crep/correction=true|false` (par défaut `true`, initialement `false`)

Cette clé, déjà rencontrée dans les exemples précédents, permet d'activer ou de désactiver le mode correctif.

`/crep/correction color`=⟨*couleur*⟩ (initialement `red`)

Cette clé permet de définir la couleur utilisée pour afficher le contenu des boîtes lorsque le mode correctif est actif.

```
\setrdcrep{correction, correction color=blue}
Question : \tcfillcrep{Réponse !}
```

Question : **Réponse!**

La couleur définie par cette clé est stockée dans la couleur `crepcorrectioncol`.

`/crep/correction font`=⟨*texte*⟩ (initialement `\bfseries\boldmath`)

Cette clé permet de définir la fonte employée pour afficher la correction.

```
\setrdcrep{correction, correction font=\sffamily
    }
Question : \tcfillcrep{Réponse !}
```

Question : Réponse !

`/crep/dotfill command`=⟨*texte*⟩ (initialement `\dotfill`)

Comme il sera vu plus bas, lorsque le mode correctif n'est pas actif, il est possible de remplir automatiquement les boîtes produites par les commandes `\tccrep` et `\tcfillcrep` (à l'aide de la clé `/tcb/crep/dotfill`). La commande de « remplissage » est alors définie par la clé `/crep/dotfill command`.

```
\setrdcrep{dotfill command=\underscorefill}
Question : \tcfillcrep[dotfill]{Réponse !}
```

Question : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Comme l'exemple précédent le prouve, l'extension également propose la commande `\underscorefill` décrite ci-dessous.

`\underscorefill`

Cette commande se comporte de la même façon que la macro `\dotfill` : elle remplit l'espace restant sur la ligne avec des tirets.
Sa définition est donnée ci-contre.

```
\newcommand\underscorefill{%
\leavevmode\xleaders%
\hbox{\_\;}\hfill\kern0pt%
}
```

`/crep/seyes=true|false` (par défaut `true`, initialement `false`)

Cette clé permet de modifier globalement le rendu des boîtes correctives générées par l'extension (pour des modifications locales, on utilisera la clé `/tcb/crep/seyes`). L'affichage avec des lignes Seyes est vu en détails dans le paragraphe **Écrire sur des lignes Seyes**.

```
\setrdcrep{seyes, correction}
Question : \tcfillcrep{Réponse !}
```

Question : **Réponse!**

Le nombre de lignes affichées dépend de la hauteur totale de la boîte qui, elle-même, dépend du contenu de celle-ci. Toutefois, modifier les valeurs des clés `/tcb/top` et/ou `/tcb/bottom`, modifie en conséquence le nombre de lignes dessinées.

```
\setrdcrep{seyes, correction}
Question : \tcfillcrep{La réponse est $\
    dfrac{2}{5}$.}
```

Question : **La réponse est $\frac{2}{5}$.**

```
\setrdcrep{seyes, correction}
Question : \tcfillcrep[top=2mm, bottom=2mm]{
    La réponse est $\dfrac{2}{5}$.}
```

Question : **La réponse est $\frac{2}{5}$.**

`/crep/seyes colors={⟨couleur⟩}{⟨couleur⟩}` (initialement `{seyescol}{seyescol}`)

Cette clé permet de modifier les couleurs utilisées par défaut pour dessiner les lignes Seyes. Le premier paramètre correspond à la couleur du trait épais tandis que le second détermine la couleur des lignes fines.

```
\setrdcrep{correction, seyes,
           seyes colors={blue}{teal}}
Question : \tcfillcrep{Réponse !}
```

Question : **Réponse!**

Les deux couleurs définies à l'aide de la clé `/crep/seyes colors` sont disponibles à travers les couleurs `seyescolA` et `seyescolB`.

`/crep/show crep=true|false` (par défaut `true`, initialement `true`)

Parfois, il peut s'avérer intéressant de désactiver l'affichage des boîtes destinées à recevoir les réponses des élèves (lorsque, par exemple, on ne souhaite que générer la consigne d'un exercice).
Attribuer la valeur `false` à la clé `/crep/show crep` entraîne les conséquences suivantes :

— la clé `/crep/correction` prend la valeur `false` ;

— les boîtes du type `crep` sont purement et simplement supprimées du rendu final ;

— les boîtes produites par les commandes `\tccrep` et `\tcfillcrep` se voient attribuer successivement les styles `normal crep style` et `tccrep show false`.

```
\tcbset{%
    normal crep style/.style={
        fontupper=\normalfont\normalcolor
    }
}
```

```
\tcbset{%
    tccrep show false/.style={%
        frame engine=empty,
        interior engine=empty,
        crep/seyes=false
    }
}
```

Autrement dit, ces boîtes deviennent transparentes et leur contenu n'est pas affiché.

— si la clé `/tcb/crep/dotfill` vaut `true`, les pointillés (ou le texte produit par la commande passée à la clé `/crep/dotfill command`) restent visibles ;

— les boîtes pour lesquelles la clé `/tcb/crep/normal crep` vaut `true` restent affichées.

```
\setrdcrep{correction, show crep}
Avant
\begin{crep}
Boîte réponse
\end{crep}

Question : \tccrep{3cm}{Réponse 1}

Autre question : \tcfillcrep{Réponse 2} *

Ou encore : \tccrep[dotfill]{3cm}{Réponse 3}


Aussi : \tcfillcrep[normal crep]{Réponse 4}
```

Avant

Boîte réponse

Question : **Réponse 1**

Autre question : **Réponse 2** *

Ou encore : **Réponse 3**

Aussi : Réponse 4

```
\setrdcrep{correction, show crep=false}
Avant
\begin{crep}
Boîte réponse
\end{crep}

Question : \tccrep{3cm}{Réponse 1}

Autre question : \tcfillcrep{Réponse 2} *

Ou encore : \tccrep[dotfill]{3cm}{Réponse 3}


Aussi : \tcfillcrep[normal crep]{Réponse 4}
```

Avant
Question :

Autre question : *

Ou encore : .....

Aussi : Réponse 4

/crep/auto factor=⟨*nombre*⟩ (initialement 2)

Cette clé affecte uniquement les boîtes produites par la commande `\tcautocrep` et modifie le calcul de la largeur de celles-ci. Lorsqu'une commande `\tcautocrep` est rencontrée, l'extension calcule la largeur naturelle du contenu de cette commande et la multiplie par ⟨*nombre*⟩. La résultat de cette opération détermine alors la largeur de la boîte produite par la commande `\tcautocrep`.

```
\setrdcrep{correction=true, auto factor=1}
Des \tcautocrep{boîtes} dont la
\tcautocrep{largeur} est automatique.
```

Des **boîtes** dont la **largeur** est automatique.

/crep/every crep (style)

/crep/every tccrep (style)

Ces deux clés permettent d'affecter un style commun à toutes les boîtes du type `crep` ou à toutes les boîtes du type `\tccrep` (plus précisément, aux boîtes produites par les commandes `\tccrep`, `\tcautocrep` et `\tcfillcrep`).

```
\setrdcrep{correction,
            every crep={
                colback=yellow,
                opacityback=0.5}
            }
\begin{crep}
Un cadre.
\end{crep}
\begin{crep}
Un autre cadre.
\end{crep}
```

Un cadre.

Un autre cadre.

```
\setrdcrep{correction,
    every tccrep={crep align=left,
        before upper=\ding{
            \numexpr201+\thetccrepcounter
            }~}
        }

Une \tcfillcrep{boîte}

Une autre \tccrep{2cm}{boîte}

Une dernière \tcautocrep{boîte}
```

Une ❶ **boîte**

Une autre ❷ **boîte**

Une dernière ❸ **boîte**

Les styles fournis par les clés `/crep/every crep` et `/crep/every tccrep` s'installent après le style défini par défaut (`crep default style` ou `tccrep default style`) et avant le style utilisateur passé en option à l'environnement ou à la commande.

`/crep/every box` (style)

Cette clé permet d'affecter un style commun aux clés `/crep/every crep` et `/crep/every tccrep`.

```
\setrdcrep{correction,
    every box={colback=yellow,
                opacityback=0.5}
        }
\begin{crep}
Un cadre.
\end{crep}
Une \tcfillcrep{boîte}
```

Un cadre.

Une **boîte**

Les clés locales (/tcb/crep)

Les clés décrites dans ce paragraphes peuvent êtres données en ⟨*option*⟩ aux boîtes correctives. Il est bien évident que ces clés, très particulières, ne sont prévues pour fonctionner qu'avec les boîtes produites par l'environnement `crep` ou qu'avec les commandes `\tccrep` et `\tcfillcrep`. C'est la raison pour laquelle ces clés ont été placées dans la sous branche **`/tcb/crep/`** de la famille **`/tcb/`**.

Pour lire leurs arguments optionnels, les commandes `\tccrep` et `\tcfillcrep` ou l'environnement `crep` parcourent successivement les branches **`/tcb/crep/`** et **`/tcb/`**. Ainsi, aucune précaution particulière n'est à prendre lorsqu'on fournit des clés en option aux commandes/environnements de l'extension.

En revanche, si l'on souhaite définir un style qui contient des clés de la branche **`/tcb/crep/`** mélangées avec des clés de la branche **`/tcb/`**, il sera nécessaire de faire attention au chemin d'accès de ces clés.

Par exemple, pour définir un style de boîte avec une couleur de fond jaune et des lignes Seyes, on pourra, au choix, écrire :

```
\setrdcrep{correction}
\tcbset{mystyle/.style={opacityback=0.25,
    colback=yellow, crep/seyes}}
\begin{crep}[mystyle]
    \lipsum[1][-2]
\end{crep}
```

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

```
\setrdcrep{correction}
\tcbset{crep/mystyle/.style={opacityback=0.25,
    colback=yellow, seyes}}
\begin{crep}[mystyle]
    \lipsum[1][-2]
\end{crep}
```

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

/tcb/crep/normal crep=true|false (par défaut `true`, initialement `false`)

Lorsque cette clé vaut `true`, le contenu de la boîte est affiché normalement (en appliquant, pour cela, le style `normal crep style`), quelle que soit la valeur de la clé `/crep/correction`.
En cas d'imbrication de boîtes correctives, les boîtes imbriquées héritent des propriétés de la boîte parent. C'est pourquoi, dans l'exemple ci-dessous, il est nécessaire de passer les options **normal crep**=false et **seyes**=false aux boîtes produites par la commande `\tccrep`.

```
\tcbset{opaq/.style={crep/seyes=false,
    normal crep=false, opacityback=0.15, colback
    =black, boxsep=3pt}}
\begin{crep}[normal crep, seyes]
On a : $(a+b)^2=\tccrep[opaq]{3cm}{a^2+2ab+b^2}$

et $(a-b)^2=\tccrep[opaq]{3cm}{a^2-2ab+b^2}$
\end{crep}
Aussi : \tcfillcrep[normal crep, seyes]{$(a+b)(a
    -b)=\tccrep[opaq]{3cm}{a^2-b^2}$}
```

On a : $(a+b)^2 =$

et $(a-b)^2 =$

Aussi : $(a+b)(a-b) =$

Dans la mesure où le style employé dans l'exemple précédent, permettant de rendre une boîte opaque, est destiné à être régulièrement utilisé, l'extension définit le style `crep opaq` de la manière suivante :

```
\tcbset{
    crep/crep opaq/.style={
        seyes=false,
        normal crep=false,
        colback=black,
        opacityback=0.15
    }
}
```

/tcb/crep/dotfill=true|false (par défaut `true`, initialement `false`)

Utilisée avec les commandes `\tccrep` ou `\tcfillcrep`, cette clé permet de remplir le cadre réponse, lorsque la correction n'est pas demandée, en utilisant la commande définie par la clé `/crep/dotfill command`.

```
\setrdcrep{dotfill command=******}
Code secret : \tccrep[dotfill]{3cm}{123456}

\setrdcrep{correction}
Code secret : \tccrep[dotfill]{3cm}{123456}
```

Code secret : ******

Code secret : **123456**

Le texte généré par la commande associée à la clé `/crep/dotfill command` est affiché avec le style `normal crep style`.

/tcb/crep/crep align=left|center|right (par défaut `center`, initialement `center`)

Cette clé permet de modifier l'alignement du contenu des boîtes produites par les commandes `\tccrep` ou `\tcfillcrep`.

```
\setrdcrep{correction}
Question : \tccrep[crep align=left]{3cm}{Réponse 1}

Question : \tcfillcrep[crep align=right]{Réponse 2}
```

Question : **Réponse 1**

Question : **Réponse 2**

/tcb/crep/crep box n (style /tcb)

Les boîtes produites à l'aide de l'environnement `crep` sont globalement numérotées (à l'intérieur de ces boîtes, le compteur est accessible par la commande `\thetcbcounter`). En modifiant le style `/tcb/crep/crep box n` (où *n* désigne le numéro de la boîte), il est possible d'affecter un style particulier à un cadre réponse donné. Ce style est appliqué après le style utilisateur fourni en option à l'environnement `crep`.

```
\setrdcrep{correction}
\tcbset{crep/crep box 11/.style={colback=yellow
    }}
\begin{crep}
Cadre \no \thetcbcounter
\end{crep}
\begin{crep}
Cadre \no \thetcbcounter
\end{crep}
```

Cadre n° 10

Cadre n° 11

/tcb/crep/tccrep box n (style /tcb)

Les boîtes produites par les commandes `\tccrep`, `\tcautocrep` et `\tcfillcrep` sont également numérotées (elles partagent le même compteur). Modifier le style référencé par `/tcb/crep/tccrep box n` (où *n* désigne le numéro de la boîte) permet d'attribuer un style spécifique à une boîte donnée. Ce style est appliqué après le style utilisateur fourni en option aux commandes citées précédemment.

```
\setrdcrep{correction}
\tcbset{crep/tccrep box 42/.style={colback=
    yellow, opacityback=0.5}}
Une boîte : \tccrep{3cm}{Boîte
                \no \thetcbcounter}

Une autre boîte : \tcfillcrep{Boîte
                    \no \thetcbcounter}
```

Une boîte : **Boîte n° 41**

Une autre boîte : **Boîte n° 42**

/tcb/crep/show crep num=true|false (par défaut `true`, initialement `false`)

Cette clé permet d'afficher le numéro interne de la boîte corrective.
Le style `crep num style`, défini ci-dessous, est alors appliqué à la boîte concernée.

```
\tcbset{%
    crep num style/.style={%
        finish={%
            \begin{tcbclipframe}
                \fill[white, opacity=0.8] (frame.north west) rectangle (frame.south east) ;
                \node[anchor=north west, fill=black!70, text=white, node font=\sffamily\tiny\bfseries]
                            at (frame.north west) {\thetcbcounter};
            \end{tcbclipframe}
        }
    }
}
```

```
\setrdcrep{correction}
Question : \tccrep[show crep num]{3cm}{Réponse}

Autre question : \tcfillcrep{Autre réponse}
```

Question : 43 Réponse

Autre question : **Autre réponse**

Pour afficher les numéros de toutes les boîtes, il suffit de modifier, en début de document, le style `crep default style` et/ou le style `tccrep default style`.

```
\setrdcrep{correction}
\tcbset{crep default style/.append style={crep/
    show crep num}}
\begin{tcbraster}[raster columns=3, raster equal
     height]
\begin{crep}
    \lipsum[1][1]
\end{crep}
\begin{crep}
    \lipsum[1][2]
\end{crep}
\begin{crep}
    \lipsum[1][3]
\end{crep}
\end{tcbraster}
```

12 Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

14 Curabitur dictum gravida mauris.

`/tcb/crep/seyes colors={`⟨*couleur*⟩`}{`⟨*couleur*⟩`}` (initialement `{seyescol}{seyescol}`)

Cette clé permet de modifier localement les couleurs utilisées par défaut pour dessiner les lignes Seyes. Le premier paramètre correspond à la couleur du trait épais tandis que le second détermine la couleur des lignes fines.

```
\setrdcrep{correction, seyes}
Question : \tcfillcrep[seyes colors=%
                {blue}{teal}]{Réponse !}
```

Question : **Réponse!**

`/tcb/crep/seyes=true|false` (par défaut `true`, initialement `false`)

Cette clé, lorsqu'on lui passe la valeur `true`, modifie l'habillage des boîtes correctives de telle sorte que des lignes au format Seyes apparaissent. Dans la mesure du possible, les paramètres d'espacement internes à la boîte sont également modifiés de façon à ce que le contenu du cadre réponse semble suivre les lignes tracées.

Lorsque cette clé est employée avec la valeur `true`, le style `crep default seyes`, défini ci-contre, vient s'ajouter au style défini par défaut pour la boîte.

```
\tcbset{%
    crep default seyes/.style={
            frame hidden, boxrule=0pt, colback=white,
            opacityback=0, sharp corners, top=1mm
    }
}
```

Il est (éventuellement) important de noter que cette clé est toujours lue en premier parmi les options fournies aux boîtes correctives quel que soit l'ordre de celles-ci (y compris lorsque la clé `/tcb/crep/seyes` apparaît au sein d'un style).

```
\setrdcrep{correction}
\tcbset{mystyle/.style={opacityback=0.25,
    colback=yellow, crep/seyes}}
\begin{crep}[mystyle]
    \lipsum[1][-2]
\end{crep}
```

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

Lorsque la clé /tcb/crep/seyes est employée avec une boîte du type crep, la commande \tcblower est redéfinie de telle sorte qu'elle produise un simple passage à la ligne. Il n'est ainsi pas possible d'obtenir une sous-boîte inférieure au sein d'une boîte crep.

```
\setrdcrep{correction}
\begin{crep}[seyes]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

/tcb/crep/forced height=true|false (par défaut true, initialement false)

En interne, l'extension rdcrep.sty possède deux mécanismes différents pour effectuer le tracé des lignes Seyes. Le mécanisme employé par défaut considère que les boîtes ont leur hauteur naturelle. Dans ce cas, il s'avère relativement aisé de déterminer comment tracer les lignes Seyes pour qu'elles soient situées sous la ligne de base du texte contenu dans la boîte. Ce mécanisme présente l'avantage de ne nécessiter qu'une seule compilation pour obtenir un tracé correct des lignes Seyes.
En revanche, dès qu'une boîte ne possède plus sa hauteur naturelle (par exemple, quand elle est modifiée par la clé /tcb/height), le mécanisme utilisé par défaut pour construire les lignes Seyes ne convient plus. Il est alors nécessaire de basculer sur un autre mode de tracé des lignes Seyes qui présente l'avantage de fonctionner même lorsque la boîte n'a pas sa hauteur naturelle mais qui présente aussi l'inconvénient de nécessiter au moins deux compilations successives avant d'obtenir un alignement correct entre les lignes et le texte.
L'extension est capable de déterminer si une boîte du type crep est utilisée au sein d'un environnement tcbraster avec la clé /tcb/raster equal height ayant pour valeur row ou none. Dans ce cas, le tracé des lignes Seyes s'effectue automatiquement avec le mode qui nécessite plusieurs compilations (ce qui, en soi, n'est pas gênant puisqu'un tcbraster avec /tcb/raster equal height différent de none demande, de toute façon, plusieurs compilations).
Cependant, de nombreuses autres circonstances peuvent amener une boîte du type crep à ne pas avoir une hauteur naturelle. Dans ce cas, l'utilisation de la clé /tcb/crep/forced height force l'extension à basculer sur le mécanisme de tracé des lignes Seyes qui nécessite plusieurs compilations successives.

```
% La boîte n'a pas une hauteur naturelle.
% On utilise donc la clé "forced height"
% (plusieurs compilations sont nécessaires)

\setrdcrep{correction}
\begin{crep}[seyes, height=4cm, forced height]
\lipsum[1][1]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

/tcb/crep/extra lines=⟨*nombre entier*⟩ (initialement 0)

Lorsque la clé /tcb/crep/seyes est utilisée conjointement avec l'environnement crep, on peut vouloir ajouter des lignes vides supplémentaires. La clé /tcb/crep/extra lines permet de préciser le nombre de lignes vides souhaitées (la valeur fournie peut-être décimale).

```
\setrdcrep{correction}
\begin{crep}[seyes, extra lines=1]
    \lipsum[1][1]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Les résultats obtenus avec la clé /tcb/crep/extra lines peuvent paraître parfois surprenants si l'on oublie

de prendre en compte (voir le paragraphe **Écrire sur des lignes Seyes**) qu'en mode Seyes, la profondeur des boîtes est annulée.
Dans l'exemple ci-dessous, la clé `/tcb/crep/extra lines` ne semble avoir aucun effet.

```
\begin{crep}[seyes, normal crep, extra lines=1]
\dembox[halign=justify, width=4cm]{Propriété
    :}{\lipsum[1][1]}
\end{crep}
```

Propriété : Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Il est pourtant « normal » qu'aucune ligne n'ait été ajoutée car, pour LaTeX, la boîte générée par la commande `\dembox` n'a pas de profondeur et la dernière ligne prise en compte est celle qui contient le texte « *Lorem ipsum dolor* ». Ainsi, pour obtenir une ligne vide sous le cadre, il aurait fallu attribuer la valeur `3` à la clé `/tcb/crep/extra lines`.

```
\begin{crep}[seyes, normal crep, extra lines=3]
\dembox[halign=justify, width=4cm]{Propriété
    :}{\lipsum[1][1]}
\end{crep}
```

Propriété : Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

On notera également que cette clé se contente d'insérer un espace vertical au sein du contenu de la boîte qui garde donc sa hauteur naturelle.

`/tcb/crep/auto factor`=⟨*nombre*⟩ (initialement 2)

Cette clé permet de modifier localement le facteur par lequel est multipliée la largeur naturelle du contenu d'une boîte produite par la commande `\tcautocrep` (voir également la clé `/crep/auto factor`).

```
\setrdcrep{correction=true}
Des \tcautocrep[auto factor=4]{boîtes} dont
la \tcautocrep[auto factor=1]{largeur} est
automatique.
```

Des **boîtes** dont la **largeur** est automatique.

Compteurs

Comme nous l'avons déjà vu avec les clés `/tcb/crep/crep box n` et `/tcb/crep/tccrep box n`, l'extension **tcolorbox.sty** fournit la commande `\thetcbcounter` qui permet d'accéder à la numérotation globale des boîtes générées par l'environnement `crep` et par les commandes `\tccrep`, `\tcautocrep` ou `\tcfillcrep`.

Cependant, parfois, il peut s'avérer intéressant de disposer d'un compteur qui numérote localement les cadres réponse.

`\thecrepcounter`

Cette commande affiche la valeur du compteur TeX sous-jacent nommé `\crepcounter` qui numérote localement (c'est-à-dire au sein du groupe courant) les différents cadres réponse produits par l'environnement `crep`.
Dans l'exemple ci-dessous, les environnements `crep` sont inclus dans deux boîtes `tcolorbox` différentes (autrement dit, dans deux groupes distincts) ce qui explique pourquoi la numérotation reprend à 1 dans le seconde boîte.

```
\setrdcrep{correction}
\tcbset{mystyle/.style={
    before upper={\textcolor{blue}{\ding{\numexpr201+\thecrepcounter}}}\space
    }
}
\begin{tcbraster}[raster columns=2, raster equal height]
\begin{tcolorbox}[colback=blue!25]
    \begin{tcbraster}[raster columns=2, raster equal height]
        \begin{crep}[mystyle]
            \lipsum[1][1]
        \end{crep}
        \begin{crep}[mystyle]
            \lipsum[1][2]
        \end{crep}
    \end{tcbraster}
\end{tcolorbox}
%
\begin{tcolorbox}[colback=blue!10]
    \begin{tcbraster}[raster columns=2, raster equal height]
        \begin{crep}[mystyle]
            \lipsum[1][1]
        \end{crep}
        \begin{crep}[mystyle]
            \lipsum[1][2]
        \end{crep}
    \end{tcbraster}
\end{tcolorbox}
\end{tcbraster}
```

❶ Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

❶ Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

`\thetccrepcounter`

De la même façon que pour la commande précédente, cette macro permet d'afficher le numéro de la boîte produite par les commandes `\tccrep`, `\tcautocrep` ou `\tcfillcrep` (ces deux types de boîtes partagent le même compteur). La numérotation est locale et le compteur TEX associé est nommé `\tccrepcounter`.

```
\setrdcrep{correction}
\tcbset{mystyle/.style={
    overlay={\node[anchor=east] at (frame.west)
        {\ding{\numexpr201+\thetccrepcounter}};}
    }
}
Écrire les réponses dans les boîtes :
\begin{itemize}[label=]
    \item \tccrep[mystyle]{3cm}{Réponse 1}
    \item \tcfillcrep[mystyle]{Réponse 2}
    \item \tcautocrep[mystyle]{Réponse 3}
\end{itemize}
```

Écrire les réponses dans les boîtes :

❶ Réponse 1

❷ Réponse 2

❸ Réponse 3

`/crep/set crep counter=`⟨*nombre entier*⟩
`/crep/set tccrep counter=`⟨*nombre entier*⟩

Ces clés permettent d'affecter une valeur donnée aux compteurs `\crepcounter` et `\tccrepcounter` (on n'ou-

bliera pas que les compteurs sont incrémentés au moment de la création de la boîte).

```
\setrdcrep{correction, set crep counter=5,
    set tccrep counter=8}
\begin{crep}
    Compteur : \thecrepcounter
\end{crep}
Et sinon : \tcfillcrep{Compteur :
        \thetccrepcounter}.
```

Compteur : 6

Et sinon : **Compteur : 9** .

/crep/reset crep counter
/crep/reset tccrep counter

Ces clés réinitialisent les compteurs `\crepcounter` et `\tccrepcounter`.
Dans l'exemple ci-dessous, il s'est avéré nécessaire d'inclure le troisième environnement `crep` à l'intérieur d'une boîte `tcolorbox` afin d'y placer également l'instruction permettant de réinitialiser le compteur (insérer cette instruction entre deux boîtes dans l'environnement `tcbraster` aurait provoqué un saut de ligne).
Cependant, le lecteur attentif remarquera que la quatrième boîte porte numéro 3 : c'est normal puisque le compteur a été réinitialisé dans un groupe (la boîte `tcolorbox`).

```
\setrdcrep{correction}
\begin{tcbraster}[raster columns=3]
    \begin{crep}
        \No \thecrepcounter
    \end{crep}
    \begin{crep}
        \No \thecrepcounter
    \end{crep}
    \begin{tcolorbox}[blankest]
        \setrdcrep{reset crep counter}%
        \begin{crep}
            \No \thecrepcounter
        \end{crep}
    \end{tcolorbox}
    \begin{crep}[raster multicolumn=3]
        \No \thecrepcounter
    \end{crep}
\end{tcbraster}
```

N° 1 **N° 2** **N° 1**

N° 3

\resetcrepcounters

Cette commande est un raccourci pour `\setrdcrep{reset crep counter, reset tccrep counter}`.

`\setcrepcounter{`⟨*nombre entier*⟩`}`
`\settccrepcounter{`⟨*nombre entier*⟩`}`

Ces commandes sont équivalentes à `\setrdcrep{set crep counter=`⟨*nombre entier*⟩`}`
et à `\setrdcrep{set tccrep counter=`⟨*nombre entier*⟩`}`.

Écrire sur des lignes Seyes

LaTeX n'est pas prévu pour mettre le texte en forme sur des lignes à écart constant. L'extension rdcrep.sty tente cependant d'adopter les meilleurs réglages permettant de faciliter la mise en forme lorsque la clé `/tcb/crep/seyes` est appliquée avec la valeur `true`. Néanmoins l'utilisateur pourra rencontrer de nombreux cas où aligner le texte avec les lignes Seyes demandera des efforts supplémentaires.

Ce paragraphe passe en revue quelques cas pratiques et propose certaines solutions destinées à éviter les écueils rencontrés.

Sauf mention contraire, les exemples qui suivent sont composés avec `\setrdcrep{correction, seyes}`.

Principes généraux

Il faut tout d'abord savoir, qu'en environnement Seyes, l'extension dessine les lignes en prenant comme point de référence la ligne de base de la dernière ligne du texte situé dans l'environnement `crep`. Ainsi, si la dernière ligne se trouve (verticalement) décalée par rapport au reste du texte, ce sont l'ensemble des lignes Seyes qui seront tracées de manière incorrecte.

❶
```
\begin{crep}
    Texte avant décalage.

    \vspace*{2mm}% décalage

    Texte après décalage.
\end{crep}
```

Texte avant décalage.

Texte après décalage.

Par ailleurs, afin de réduire les risques de décalage, l'extension rdcrep.sty fait en sorte qu'en mode Seyes, la hauteur et la profondeur des différents éléments n'interfèrent pas dans le calcul de l'espace entre les lignes.

❷
```
\begin{crep}
Début : \vtop{a\\ b\\ c}
\end{crep}
```

Début : a
b
c

❸
```
\begin{crep}
Début : \vtop{a\\ b\\ c}

Fin.
\end{crep}
```

Début : a
Fin. b
c

Dans l'exemple ❷, ci-dessus, la dernière ligne du paragraphe contenu dans l'environnement `crep` a une profondeur très grande, prise en compte (car c'est la dernière ligne) pour le dessin des lignes horizontales.

En revanche, dans l'exemple ❸, la dernière ligne est celle qui contient le texte « fin ». L'extension se base donc sur cette ligne pour tracer les lignes horizontales. De surcroît, la profondeur de l'objet `\vtop{a\\ b\\ c}`, sur la première ligne, n'est pas prise en compte. Ces deux raisons expliquent, d'une part, pourquoi le texte « fin. » n'est pas situé sous le texte « c » et, d'autre part, l'absence de ligne tracée au niveau de la lettre « c ».

Ainsi, si l'on souhaite que le texte « fin » soit situé sous la lettre « c », il sera nécessaire de sauter des lignes. Pour cela, on peut se servir d'une instruction du type `\vspace*{2\baselineskip}` ou encore recourir à la commande `\seyesskip`.

`\seyesskip`[⟨*nombre*⟩]

Cette commande est un raccourci pour `\vspace*{`⟨*nombre*⟩`\baselineskip}`.
Le paramètre ⟨*nombre*⟩ peut être omis et, dans ce cas, une seule ligne est sautée.

```
\begin{crep}
Début : \vtop{a\\ b\\ c}

\seyesskip[2]

Fin.
\end{crep}
```

Début : a
b
c

Fin.

Dans le même esprit, la hauteur des différents éléments n'est prise en compte que lorsque ceux-ci sont situés sur la première ligne du texte contenu dans l'environnement `crep`.

❹
```
\begin{crep}
Avant : \vbox{a\\ b\\ c}
\end{crep}
```

a
b
Avant : c

❺
```
\begin{crep}
Avant :

Après : \vbox{a\\ b\\ c}
\end{crep}
```

a
Avant : b
Après : c

Dans l'exemple ❹, l'élément `\vbox{a\\ b\\ c}` est situé sur la première ligne, sa hauteur est donc prise en compte et les lignes horizontales sont correctement tracées.

En revanche, dans l'exemple ❺, le même élément est situé sur la deuxième ligne et, par conséquent, sa hauteur ne vient pas perturber l'espacement entre les lignes. En contrepartie, puisque l'environnement `crep` ne tient pas compte de la hauteur de cet élément, la boîte verticale générée par `\vbox{a\\ b\\ c}` « déborde » du cadre `crep`. Là encore, il sera nécessaire d'introduire des sauts de lignes pour obtenir une mise en forme satisfaisante.

Une dernière remarque : pour obtenir un alignement correct le long des lignes horizontales, il s'avère nécessaire d'activer (lorsqu'elle est disponible) l'option `[t]` des environnements utilisés.

❻
```
\begin{crep}
\begin{minipage}{0.4\linewidth}
    \lipsum[1][-2]
\end{minipage}%
\hfill%
\begin{minipage}{0.4\linewidth}
    \lipsum[1][3-4]
\end{minipage}%
\end{crep}
```

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

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

❼
```
\begin{crep}
\begin{minipage}[t]{0.4\linewidth}
    \lipsum[1][-2]
\end{minipage}%
\hfill%
\begin{minipage}[t]{0.4\linewidth}
    \lipsum[1][3-4]
\end{minipage}%
\end{crep}
```

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

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

Dans l'exemple ❻, l'option `[t]` n'a pas été utilisée et le texte de la première colonne est légèrement décalé vers le bas tandis que celui de la seconde colonne présente un décalage vertical vers le haut.

Avec l'option `[t]` (exemple ❼), l'ensemble du texte est correctement aligné le long des lignes Seyes.

Mathématiques

Observons les trois exemples ci-dessous :

❶
```
\begin{crep}[extra lines=1]
Texte avant :
\[ (3x)^2 = 9x^2 \]
Texte après
\end{crep}
```

Texte avant :

$$(3x)^2 = 9x^2$$

Texte après

❷
```
\begin{crep}
Texte avant :

\[ (3x)^2 = 9x^2 \]
Texte après
\end{crep}
```

Texte avant :

$$(3x)^2 = 9x^2$$

Texte après

❸
```
\begin{crep}[extra lines=2]
Texte avant : $\dfrac{1}
{\dfrac{2}{3}}$

Texte après
\end{crep}
```

Texte avant : $\dfrac{1}{\dfrac{2}{3}}$

Texte après

L'exemple ❶ montre que le passage en mode mathématiques à l'aide de `\[...\]` reste fonctionnel lorsque la clé `/tcb/crep/seyes` vaut `true`. En revanche (exemple ❷), une ligne vide dans le code source avant l'utilisation de `\[...\]` entraîne un saut de ligne dans le résultat final.

L'exemple ❸ illustre un mécanisme général du mode Seyes : les lignes dont la profondeur dépasse 8 mm ne sont pas prises en compte dans la gestion de l'espacement vertical. Dans ces cas là, l'utilisateur sera contraint d'introduire des sauts de lignes manuels à l'aide, par exemple, de la commande `\seyesskip` étudiée précédemment.

La plupart des environnements permettant des alignements sont utilisables en mode Seyes.

❹
```
\begin{crep}
Texte avant :
\begin{align*}
    a&=b+c  \\
    d&=e+f
\end{align*}
Texte après
\end{crep}
```

Texte avant :

$$a = b + c$$

$$d = e + f$$

Texte après

❺
```
\begin{crep}
Texte avant :
\begin{gather*}
    a=b+c  \\
    d=e+f
\end{gather*}
Texte après
\end{crep}
```

Texte avant :

$$a = b + c$$

$$d = e + f$$

Texte après

❻
```
\begin{crep}[extra lines=2]
Texte avant : %
$\begin{aligned}[t]
    a&=b+c  \\
    d&=e+f
\end{aligned}$

Un long texte après
\end{crep}
```

Texte avant : $a = b + c$

Un long texte après $d = e + f$

Les exemples ❹ et ❺ permettent de constater que la majorité des environnements d'alignements proposés par le package **amsmath.sty** (`align`, `alignat`, `xalignat`, `flalign`, `multiline`, ...) restent compatibles avec le mode Seyes.

En revanche (exemple ❻), le recours aux environnements du type `aligned` ou `gathered` nécessite de la part de l'utilisateur qu'il introduise manuellement un ou plusieurs sauts de lignes après ces environnements. Il faudra également veiller à utiliser l'option `[t]` avec ces environnements pour un alignement correct le long des lignes.

L'utilisation des tableaux du type `array` s'avère plus problématique :

❼
```
\begin{crep}[extra lines=1]
Texte avant :
\[
\begin{array}[t]{rcl}
    a&= &b+c  \\
    d&=& e+f
\end{array}
\]
Texte après
\end{crep}
```

Texte avant :

$$\begin{array}[t]{rcl} a &= & b+c \\ d &=& e+f \end{array}$$

Texte après

❽
```
\begin{crep}
Texte avant :
\setlength{\arrayrulewidth
    }{3pt}
\[
\begin{array}[t]{rcl}
    a&= &b+c  \\ \hline
    d&=& e+f
\end{array}
\]

\seyesskip

Texte après
\end{crep}
```

Texte avant :

$$\begin{array}[t]{rcl} a &= & b+c \\ \hline d &=& e+f \end{array}$$

Texte après

❾
```
\begin{crep}
Texte avant :
\[
\left\{
\begin{array}[t]{rcl}
    a&= &b+c  \\
    d&=& e+f
\end{array}\right.
\]

\seyesskip

Texte après
\end{crep}
```

Texte avant :

$$\left\{ \begin{array}[t]{rcl} a &= & b+c \\ d &=& e+f \end{array}\right.$$

Texte après

L'exemple ❼ montre qu'il s'avère nécessaire d'introduire des sauts de lignes manuels après l'utilisation d'une

structure du type `array`.

Dans l'exemple ❽, on constate que l'épaisseur de la ligne horizontale décale le texte de la ligne suivante vers le bas. L'exemple ❾, quant à lui, prouve que la taille des délimiteurs n'est pas correctement calculée.

Ces problèmes peuvent toutefois être résolus (voir le sous-paragraphe **Tableaux**).

Listes

Les environnements de mise en forme de listes sont utilisables en mode Seyes à condition de charger l'extension **enumitem.sty** et d'utiliser l'option `nosep` avec ces listes.

❶
```
\begin{crep}
\begin{enumerate}[nosep]
    \item Item 1
    \item %
        \begin{enumerate}[
    nosep]
            \item Item 2-1
            \item Item 2-2
        \end{enumerate}
    \item Item 3
\end{enumerate}
\end{crep}
```

1. Item 1
2. (a) Item 2-1
   (b) Item 2-2
3. Item 3

❷
```
\begin{crep}
\begin{itemize}[nosep]
    \item Item 1
    \item Item 2 un peu plus
     long que les autres
    \item Item 3
\end{itemize}
\end{crep}
```

— Item 1
— Item 2 un peu plus long que les autres
— Item 3

❸
```
\begin{crep}
\begin{description}[nosep]
    \item[A :] Item 1
    \item[B :] %
        \begin{itemize}[
    nosep]
            \item Item 2-1
            \item Item 2-2
        \end{itemize}
    \item[C :] Item 3
\end{description}
\end{crep}
```

A : Item 1
B : — Item 2-1
— Item 2-2
C : Item 3

Tableaux

L'insertion de tableaux, en mode Seyes, s'avère délicate mais reste néanmoins possible pour des tableaux de faible complexité.

Lorsqu'un tableau ne contient aucun filet horizontal, le recours aux environnements habituels est possible à condition de les inclure dans un groupe contenant la commande `\initSeyesTab` en début de groupe.

`\initSeyesTab`

Cette commande permet de régler quelques paramètres destinés à produire un affichage convenable des tableaux en mode Seyes. On utilisera cette commande au sein d'un groupe afin que l'application des réglages reste locale.

```
\begin{crep}
Un tableau sans filet : {\initSeyesTab%
\begin{tabular}[t]{lcc}
Ligne 1 & a & b  \\
Ligne 2 & c & d
\end{tabular}}
\end{crep}
```

Un tableau sans filet : Ligne 1 a b
Ligne 2 c d

Dans tous les cas, l'insertion d'un tableau s'avérera difficile si l'une de ses cellules possède une profondeur (ou une hauteur) trop importante. En effet, un décalage des lignes suivantes du tableau vers le bas sera alors introduit (exemple ❶ ci-dessous).

❶
```
\begin{crep}
Tableau : {\initSeyesTab%
\begin{tabular}[t]{lcc}
Ligne 1 & a & $\dfrac{1}{\dfrac{2}{3}}$  \\
Ligne 2 & c & d
\end{tabular}}
\end{crep}
```

Tableau : Ligne 1 a $\dfrac{1}{\dfrac{2}{3}}$

Ligne 2 c d

❷
```
\begin{crep}
Tableau : {\initSeyesTab%
\begin{tabular}[t]{lcc}
Ligne 1 & a &
    \smash[b]{$\dfrac{1}{\dfrac{2}{3}}$}  \\
 & &    \\
Ligne 2 & c & d
\end{tabular}}
\end{crep}
```

Tableau : Ligne 1 a $\dfrac{1}{\dfrac{2}{3}}$

Ligne 2 c d

L'exemple ❷ montre qu'il est toutefois possible de contourner l'obstacle en annulant la profondeur de la cellule à l'aide d'une instruction `\smash[b]{...}` et en ajoutant une ligne vide au tableau.

Construire des tableaux avec des filets horizontaux est permis à condition de recourir aux outils proposés par l'extension **nicematrix.sty**. En effet, cette extension permet d'insérer des filets horizontaux dont l'épaisseur ne vient pas perturber l'alignement vertical des cellules. Il sera toutefois nécessaire de se servir de la commande `\sline` afin de tracer des filets horizontaux et, de par le fonctionnement de **nicematrix.sty**, plusieurs compilations successives seront nécessaires avant d'obtenir le tableau final.

\sline

Cette commande, qui permet de tracer des filets horizontaux, n'est utilisable qu'au sein des environnements fournis par l'extension **nicematrix.sty** et uniquement lorsque le tableau est inclus dans un groupe contenant l'instruction `\initSeyesTab`.
En effet, la commande `\initSeyesTab` permet (entre autres) de déclarer un nouveau type de filet dessiné grâce à Ti*k*Z (voir ci-contre).

```
\NiceMatrixOptions{%
    standard-cline,
    custom-line =
        {
        letter = I ,
        tikz = {line width=1.0975*\arrayrulewidth},
        command = sline ,
        }
}
```

```
\begin{crep}
Un tableau : {\initSeyesTab%
\begin{NiceTabular}[t]{lcc}%
            [vlines, rules/width=1pt]
\sline
Ligne 1 & a & b  \\ \sline
Ligne 2 & c & d \\ \sline
\end{NiceTabular}}
\end{crep}
```

Un tableau :

Ligne 1	a	b
Ligne 2	c	d

Pour tracer un filet horizontal partiel, on aura recours à la commande `\cline` :

```
\begin{crep}
Un tableau : {\initSeyesTab%
\begin{NiceTabular}[t]{lcc}%
            [vlines, rules/width=1pt]
\sline
Ligne 1 & a & b  \\ \cline{1-1}
Ligne 2 & c & d \\ \sline
\end{NiceTabular}}
\end{crep}
```

Un tableau : | Ligne 1 | a | b |
|---|---|---|
| Ligne 2 | c | d |

Remarque : en théorie, dans l'exemple précédent, on aurait pu écrire `\cline{1}` au lieu de `\cline{1-1}`. Pourtant, en pratique, l'usage de `\cline{1}` provoque une erreur de compilation.

L'extension rdcrep.sty propose quelques environnements qui permettent de simplifier la saisie.

`\begin{seyestab}{`⟨*spécifications de colonnes*⟩`}[`⟨*options*⟩`]`
⟨*contenu d'environnement*⟩
`\end{seyestab}`

En mode Seyes, cet environnement remplace l'environnement `NiceTabular`.

```
\begin{crep}
Avant :
\begin{center}
\begin{seyestab}{lcc}[vlines, rules/width=1pt]
    \CodeBefore
        \columncolor[opacity=0.5]{red!15}{1}
    \Body
    \sline
    Ligne 1 & a & b \\ \sline
    Ligne 2 & c & d \\ \sline
    Ligne 3 & e & f \\ \sline
\end{seyestab}
\end{center}

\seyesskip[2]

Après.
\end{crep}
```

Avant :

Ligne 1	a	b
Ligne 2	c	d
Ligne 3	e	f

Après.

`\begin{seyestabx}{`⟨*longueur*⟩`}{`⟨*spécifications de colonnes*⟩`}[`⟨*options*⟩`]`
⟨*contenu d'environnement*⟩
`\end{seyestabx}`

En mode Seyes, cet environnement remplace l'environnement `NiceTabularX`.

```
\begin{crep}
Avant :
\begin{center}
\begin{seyestabx}{0.8\linewidth}{XX[c]X[c]}
                    [vlines, rules/width=1pt]
    \CodeBefore
        \columncolor[opacity=0.5]{red!15}{1}
    \Body
    \sline
    Ligne 1 & a & b \\ \sline
    Ligne 2 & c & d \\ \sline
    Ligne 3 & e & f \\ \sline
\end{seyestabx}
\end{center}

\seyesskip[2]

Après.
\end{crep}
```

Avant :

Ligne 1	**a**	**b**
Ligne 2	**c**	**d**
Ligne 3	**e**	**f**

Après.

`\begin{seyesarray}{`⟨*spécifications de colonnes*⟩`}[`⟨*options*⟩`]`
⟨*contenu d'environnement*⟩
`\end{seyesarray}`

En mode Seyes, cet environnement remplace l'environnement `NiceArray`.

```
\begin{crep}
Avant :
\[
\begin{seyesarray}{rcl}
    f(x) &= & 2x-1  \\
    g(x) &= & -3x+5
\end{seyesarray}
\]

\seyesskip

Après.
\end{crep}
```

Avant :

$$f(x) = 2x-1$$
$$g(x) = -3x+5$$

Après.

`\begin{seyesarraydelims}{`⟨*délimiteur ouvrant*⟩`}{`⟨*délimiteur fermant*⟩`}{`⟨*spécifications de col.*⟩`}[`⟨*options*⟩`]`
⟨*contenu d'environnement*⟩
`\end{seyesarraydelims}`

En mode Seyes, cet environnement remplace l'environnement `NiceArrayWithDelims`.

```
\begin{crep}
Avant :
\[
\begin{seyesarraydelims}{\{}{.}{l}
    f(x) =  2x-1  \\
    g(x) =  -3x+5
\end{seyesarraydelims}
\]

\seyesskip

Après.
\end{crep}
```

Avant :

$$\begin{cases} f(x) = 2x-1 \\ g(x) = -3x+5 \end{cases}$$

Après.

Propriétés mathématiques

Afin de mettre en forme la rédaction de pas de raisonnement, l'extension rdcrep.sty propose la commande \dembox conçue pour être utilisée au sein d'un environnement crep.

\dembox[⟨*options*⟩]{⟨*texte*⟩}{⟨*texte*⟩}

Cette commande produit une boîte du type tcbox pourvue d'un habillage spécifique.
Le premier paramètre obligatoire désigne le titre de la boîte tandis que le second permet de spécifier le contenu de la boîte.

```
\setrdcrep{correction}
\begin{crep}[seyes]
\dembox[width=3cm]{Ce que je sais :}{$(HS) \perp (HT)$\\$(AT) \perp (HT)$}%
\hfill%
\dembox[width=8cm, halign=justify]{Propriété :}{Si deux droites sont perpendiculaires à une
    même troisième droite alors ces deux droites sont parallèles entre elles.}%
\hfill%
\dembox[width=4cm]{Conclusion :}{$(AT) \parallelslant (HS)$}
\end{crep}
```

Ce que je sais :
$(HS) \perp (HT)$
$(AT) \perp (HT)$

Propriété :
Si deux droites sont perpendiculaires à une même troisième droite alors ces deux droites sont parallèles entre elles.

Conclusion :
$(AT) \parallelslant (HS)$

Par défaut, le contenu des boîtes est centré. On utilisera la clé /tcb/halign pour modifier l'alignement.
Lorsque la clé /tcb/width n'est pas fournie, la boîte produite s'étend sur toute la largeur disponible.

```
\setrdcrep{correction}
\begin{crep}[seyes]
\dembox[halign=justify]{Propriété :}{Si
    deux droites sont perpendiculaires à
    une même troisième droite alors ces
    deux droites sont parallèles entre
    elles.}
\end{crep}
```

Propriété :
Si deux droites sont perpendiculaires à une même troisième droite alors ces deux droites sont parallèles entre elles.

La commande \dembox détecte si l'environnement crep englobant est utilisé ou non avec une valeur true pour la clé /tcb/crep/normal crep. Lorsque **normal crep**=true, le style de la boîte est automatiquement modifié.

```
\setrdcrep{correction}
\begin{crep}[seyes, normal crep]
\dembox[halign=justify]{Propriété :}{Si
    deux droites sont perpendiculaires à
    une même troisième droite alors ces
    deux droites sont parallèles entre
    elles.}
\end{crep}
```

Propriété :
Si deux droites sont perpendiculaires à une même troisième droite alors ces deux droites sont parallèles entre elles.

/tcb/dembox/demskin=⟨*nom d'un habillage*⟩ (initialement regular)

Cette clé permet de modifier l'habillage de la boîte produite par la commande \dembox. Par défaut, seules deux valeurs sont possibles (regular et compact) mais les habillages disponibles peuvent être étendus par l'utilisateur.

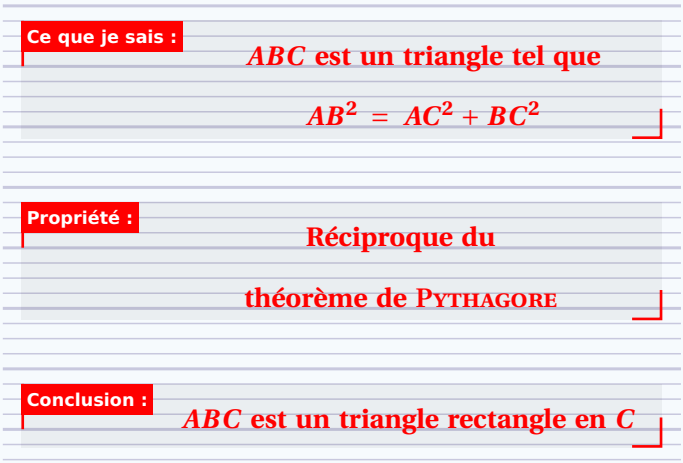
```
\setrdcrep{correction}
\begin{crep}[seyes]
\dembox[demskin=compact]{Ce que
    je sais :}{$ABC$ est un
    triangle tel que\\
    $AB^2 = AC^2 + BC^2$}

\seyesskip[2]

\dembox[demskin=compact]{
    Propriété :}{Réciproque du\\
    théorème de \bsc{Pythagore}}

\seyesskip[2]

\dembox[demskin=compact]{
    Conclusion :}{$ABC$ est un
    triangle rectangle en $C$}
\end{crep}
```



L'habillage des boîtes produites par la commande `\dembox` peut, dans une certaine mesure, être personnalisé. En effet, l'extension applique, dans cet ordre, les styles suivants à la boîte `tcbox` :

— `dembox correction` (si `normal crep`=false) ou `dembox normal crep` (si `normal crep`=true) ;

— `dembox title` (pour la mise en forme du titre de la boîte) ;

— `dembox style`.

De surcroît, il faut savoir que ces trois styles doivent appartenir à une sous-branche de la branche `/tcb/dembox/` nommée de manière identique à l'habillage.

La page suivante montre, par exemple, les styles utilisés pour les deux habillages `regular` et `compact`.

Ainsi, si l'on souhaite créer un nouvel habillage, nommé, par exemple, *monskin*, pour les boîtes générées par la commande `\dembox`, la définition des styles aura la structure ci-contre.

On veillera à encadrer l'instruction `\tcbset{...}` des commandes `\makeatletter` et `\makeatother` si l'on doit recourir à des macros privées dans les différentes définitions de styles.

Pour convoquer ce nouvel habillage, on écrira simplement `demskin=monskin` parmi les options de la commande `\dembox`.

```
\tcbset{%
    dembox/monskin/.cd,
    dembox correction/.style={
        ...
    },
    dembox normal crep/.style={
        ...
    },
    dembox style/.style={
        ...
    },
    dembox title/.style={
        ...
    }
}
```

Instructions conditionnelles

Les instructions qui suivent permettent une mise en forme conditionnelle. Elles peuvent également être utilisées au sein de macros personnelles.

`\SiCorrection`{⟨*texte*⟩}
`\SiCorrection*`{⟨*texte*⟩}

L'argument de cette commande n'est traité que lorsque la clé `/crep/correction` vaut `true`.

Habillage *Regular*

```
\tcbset{%
    dembox/regular/.cd,
    dembox correction/.style={
        colframe=crepcorrectioncol,
        coltext=crepcorrectioncol,
    },
    dembox normal crep/.style={
        colframe=black, coltext=black
    },
    dembox style/.style={%
        enhanced jigsaw, opacityback=0.05,
        colback=black, sharp corners,
        frame hidden, boxsep=0pt, boxrule=1pt,
        top=8pt, bottom=4pt, left=6pt, right=6pt,
        interior code={%
            \fill[tcbcolback,
                fill opacity=\kvtcb@opacityback]
                (frame.north west)
                    rectangle (frame.south east);
        },
        finish={%
            \draw[tcbcolframe,
                line width=\kvtcb@left@rule]
            ([shift={(-4mm,0.5*\pgflinewidth)}]
                frame.south east) -|
            ([shift={(-0.5*\pgflinewidth,4mm)}]
                frame.south east) ;
            \draw[tcbcolframe,
                line width=\kvtcb@left@rule]
                ([xshift=0.5*\pgflinewidth]
                frame.north west) --++(0,-4mm) ;
        }
    },
    dembox title/.style={%
        adjusted title=#1,
        attach boxed title to top left,
        fonttitle=\sffamily\bfseries\scriptsize,
        boxed title style={
            sharp corners, frame hidden,
            boxrule=0pt, colback=tcbcolframe,
            boxsep=2pt, left=0pt,
            right=0pt, top=0pt, bottom=0pt
        }
    }
}
```

Habillage *Compact*

```
\tcbset{%
    dembox/compact/.cd,
    dembox compact title width/.code={
        \setbox0\hbox{\tcbtitle}%
        \tcbdimto\demboxcompacttitlewidth%
                                {\wd0+4pt+6pt}
    },
    dembox correction/.style={
        colframe=crepcorrectioncol,
        coltext=crepcorrectioncol,
    },
    dembox normal crep/.style={
        colframe=black, coltext=black
    },
    dembox style/.style={%
        compact/dembox compact title width,
        enhanced jigsaw, opacityback=0.05,
        colback=black, sharp corners,
        frame hidden, boxsep=0pt,
        top=8pt, bottom=4pt, left=6pt, right=6pt,
        boxrule=1pt,
        left=\demboxcompacttitlewidth,
        interior code={%
            \fill[tcbcolback,
                fill opacity=\kvtcb@opacityback]
            (frame.north west) rectangle
                            (frame.south east);
        },
        finish={%
            \draw[tcbcolframe,
                line width=\kvtcb@left@rule]
            ([shift={(-4mm,0.5*\pgflinewidth)}]
                frame.south east) -|
            ([shift={(-0.5*\pgflinewidth,4mm)}]
                frame.south east) ;
            \draw[tcbcolframe,
                line width=\kvtcb@left@rule]
            ([xshift=0.5*\pgflinewidth]
                frame.north west) --++(0,-6mm) ;
        }
    },
    dembox title/.style={%
        adjusted title=#1,
        attach boxed title to top left={
                yshift=-\tcboxedtitleheight},
        fonttitle=\sffamily\bfseries\scriptsize,
        boxed title style={
            sharp corners, frame hidden,
            boxrule=0pt, colback=tcbcolframe,
            boxsep=2pt, left=0pt,
            right=0pt, top=0pt, bottom=0pt
        }
    }
}
```

```
\setrdcrep{correction=false}
\SiCorrection{[Mode correctif]} Avant

\SiCorrection{Nouveau paragraphe.

Qui n'apparaît que si correction=true.}

Après
```

Avant
Après

```
\setrdcrep{correction=true}
\SiCorrection{[Mode correctif]} Avant

\SiCorrection{Nouveau paragraphe.

Qui n'apparaît que si correction=true.}

Après
```

[Mode correctif] Avant
Nouveau paragraphe.
Qui n'apparaît que si correction=true.
Après

La version étoilée de cette commande prend en compte la valeur des clés `/crep/correction color` et `/crep/correction font` pour afficher ⟨*texte*⟩ lorsque la clé `/crep/correction` vaut `true`.

```
\setrdcrep{correction=false}
Avant

\SiCorrection*{Nouveau paragraphe.

Qui n'apparaît que si correction=true.}

Après
```

Avant
Après

```
\setrdcrep{correction=true}
Avant

\SiCorrection*{Nouveau paragraphe.

Qui n'apparaît que si correction=true.}

Après
```

Avant
Nouveau paragraphe.
Qui n'apparaît que si correction=true.
Après

`\TFCorrection`{⟨*texte si vrai*⟩}{⟨*texte si false*⟩}

Cette commande prend deux arguments. Le premier n'est traité qu'à la condition où la clé `/crep/correction` vaut `true` tandis qu'au contraire, le second n'est traité que lorsque `correction=false`.

```
\setrdcrep{correction=false}
Avant

\TFCorrection{\begin{tcolorbox}}%
                    {\begin{center}}
Paragraphe.

Fin paragraphe.
\TFCorrection{\end{tcolorbox}}{\end{center}}


Après
```

Avant

Paragraphe.
Fin paragraphe.

Après

```
\setrdcrep{correction=true}
Avant

\TFCorrection{\begin{tcolorbox}}%
                    {\begin{center}}
Paragraphe.

Fin paragraphe.
\TFCorrection{\end{tcolorbox}}{\end{center}}


Après
```

Avant

Paragraphe.
Fin paragraphe.

Après

`\SiCrep{`⟨*texte*⟩`}`

L'argument de cette commande n'est traité que lorsque la clé `/crep/show crep` vaut `true`.

```
\setrdcrep{show crep=true}
Quel est l'âge du capitaine ?
\begin{crep}
Le capitaine a un âge canonique !
\end{crep}
\SiCrep{Inscrire votre réponse dans le
    cadre ci-dessus}
```

Quel est l'âge du capitaine?

Le capitaine a un âge canonique!

Inscrire votre réponse dans le cadre ci-dessus

```
\setrdcrep{show crep=false}
Quel est l'âge du capitaine ?
\begin{crep}
Le capitaine a un âge canonique !
\end{crep}
\SiCrep{Inscrire votre réponse dans le
    cadre ci-dessus}
```

Quel est l'âge du capitaine?

`\TFCrep{`⟨*texte si vrai*⟩`}{`⟨*texte si false*⟩`}`

Cette commande prend deux arguments. Le premier n'est traité qu'à la condition où la clé `/crep/show crep` vaut `true` tandis qu'au contraire, le second n'est traité que lorsque `show crep=false`.

```
\setrdcrep{show crep=false}

\TFCrep{Développer $A$ ci-dessous :}
        {Développer $A$ sur votre copie.}

$A=(2x+1)(5-3x)\SiCrep{=}\tcfillcrep{10x-6x
    ^2+5-3x}$
```

Développer A sur votre copie.

$A = (2x+1)(5-3x)$

```
\setrdcrep{show crep=true}

\TFCrep{Développer $A$ ci-dessous :}
        {Développer $A$ sur votre copie.}

$A=(2x+1)(5-3x)\SiCrep{=}\tcfillcrep{10x-6x
    ^2+5-3x}$
```

Développer A ci-dessous :

$A = (2x+1)(5-3x) =$ $\mathbf{10x-6x^2+5-3x}$

Outre les commandes précédentes, l'extension propose également un nouvel « handler » permettant d'obtenir

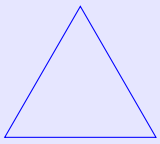
des styles différents selon la valeur de la clé `/crep/correction`.

`/handlers/.cor style`={⟨*style si vrai*⟩}{⟨*style si faux*⟩}

Cet *handler* s'utilise de la même façon que `.style` et permet de créer un style dont la définition est donnée par le premier argument lorsque `/crep/correction` vaut `true` et par le second sinon. Cet *handler* s'utilise avec n'importe quelle famille (en particulier, aussi bien `/tcb/` que `/tikz/`).

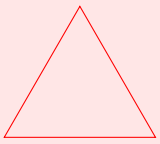
```
\setrdcrep{correction=false}
\tcbset{
    mytcbstyle/.cor style={colback=red!10,
    colframe=red}{colback=blue!10, colframe
    =blue}
}
\tikzset{
    mytikzstyle/.cor style={draw=red}{draw=
    blue}
}
\begin{tcolorbox}[mytcbstyle]
Une jolie figure :
\begin{center}
    \begin{tikzpicture}
        \draw[mytikzstyle] (0,0)--++(2,0)
                           --++(120:2)--cycle;
    \end{tikzpicture}
\end{center}
\end{tcolorbox}
```

Une jolie figure :



```
\setrdcrep{correction=true}
\tcbset{
    mytcbstyle/.cor style={colback=red!10,
    colframe=red}{colback=blue!10, colframe
    =blue}
}
\tikzset{
    mytikzstyle/.cor style={draw=red}{draw=
    blue}
}
\begin{tcolorbox}[mytcbstyle]
Une jolie figure :
\begin{center}
    \begin{tikzpicture}
        \draw[mytikzstyle] (0,0)--++(2,0)
                           --++(120:2)--cycle;
    \end{tikzpicture}
\end{center}
\end{tcolorbox}
```

Une jolie figure :



Boîtes dérivées

Dans certaines circonstances, il peut s'avérer intéressant de disposer d'une boîte constituée de deux « parties » dont l'une est sensible au mode correctif et l'autre non.

`/tcb/crep/crep mode`=`regular|upper|lower|left|right` (par défaut `regular`, initialement `regular`)

La boîte produite par défaut avec l'environnement `crep` correspond à la valeur `regular` attribuée à la clé `/tcb/crep/crep mode` (et, dans ce cas, on rappelle que la commande `\tcblower` entraîne un simple retour à la ligne).

En revanche, avec les autres valeurs possibles, la boîte produite possède à la fois une partie supérieure et une partie inférieure séparées par la commande `\tcblower`. La valeur de la clé indique la position de la partie de la boîte sensible au mode correctif.

Lorsque les valeurs `left` ou `right` sont employées, le style `/tcb/sidebyside` est automatiquement activé. De surcroît, la valeur de la clé `/tcb/sidebyside gap` est augmentée du double de la valeur de la clé `/tcb/boxsep` de telle sorte que les contenus des deux boîtes internes ne soient pas accolés l'un contre l'autre lorsque `sidebyside gap=0pt` (par défaut).

```
\begin{crep}[crep mode=upper]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

```
\begin{crep}[crep mode=lower]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

```
\begin{crep}[crep mode=left]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

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

```
\begin{crep}[crep mode=right]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

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

Il est, bien évidemment, possible d'activer la clé `/tcb/crep/seyes` avec l'un des modes disponibles. Cependant, avec les valeurs `left` et `right`, le mécanisme interne chargé de tracer les lignes Seyes est modifié afin de rester compatible avec l'utilisation de la clé `/tcb/sidebyside align` et deux compilations successives seront nécessaires avant d'obtenir un tracé correct de ces lignes.

```
\begin{crep}[seyes, crep mode=upper]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

```
\begin{crep}[seyes, crep mode=lower]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

```
\begin{crep}[seyes, crep mode=left]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

```
\begin{crep}[seyes, crep mode=right]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

Par défaut, le passage au mode Seyes désactive le tracé du cadre autour de la boîte. En cas de besoin, il est aisé de rétablir son tracé en modifiant le style du cadre réponse.

```
\begin{crep}[seyes, crep mode=right,
    frame style={}, boxrule=0.8pt,
    rounded corners, arc=3mm,
    colframe=seyescolA]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

`/tcb/crep/seyes upper` (style)
`/tcb/crep/seyes lower` (style)
`/tcb/crep/seyes left` (style)
`/tcb/crep/seyes right` (style)

Ces clés définissent quatre styles qui s'utilisent comme des raccourcis :

```
\tcbset{
crep/.cd,
seyes upper/.style={
    seyes,
    crep mode=upper
    }
}
```

```
\tcbset{
crep/.cd,
seyes lower/.style={
    seyes,
    crep mode=lower
    }
}
```

```
\tcbset{
crep/.cd,
seyes left/.style={
    seyes,
    crep mode=left
    }
}
```

```
\tcbset{
crep/.cd,
seyes upper/.style={
    seyes,
    crep mode=right
    }
}
```

```
\begin{crep}[seyes left,
    extra lines=2, top=1.5mm,
    bottom=1mm, lefthand width=4
    cm, sidebyside align=top,
    frame style={}, boxrule=0.8pt
    , rounded corners, arc=3mm,
    colframe=seyescolA]
    \lipsum[1][1]
\tcblower
    \lipsum[1][2]
\end{crep}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

La clé /tcb/bbox

`/tcb/bbox`={⟨*longueur*⟩}{⟨*longueur*⟩} (initialement {0mm}{0mm})

Cette clé est un simple style défini de la façon suivante :

```
\tcbset{%
    bbox/.style 2 args={enlarge bottom by=#1, enlarge top by=#2}
}
```

Grâce à cette clé, il est ainsi possible d'augmenter aisément l'espace vertical en-dessous et au-dessus d'une boîte du type `tcolorbox` ou `tcbox`. Dans le cadre de cette extension, cela permet, si besoin, de laisser davantage de place que prévu par défaut lors de l'utilisation des commandes `\tccrep` ou `\tcfillcrep`.

```
\setrdcrep{correction}
\begin{tabularx}{\linewidth}{XX}
Un petit paragraphe pour
\tccrep[seyes]{2cm}{illustrer} la
situation.
&
Un petit paragraphe pour
\tccrep[seyes,
        bbox={2mm}{2mm}]{2cm}{illustrer}
la situation.
\end{tabularx}
```

Un petit paragraphe pour illustrer la situation.

Un petit paragraphe pour illustrer la situation.

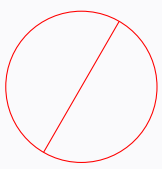
Le style seyesfig

Cet habillage des boîtes du type `crep` peut être destiné à proposer aux élèves un encadré dans lequel construire leurs figures.
La définition du style est donnée ci-contre.

```
\tcbset{%
    seyesfig/.style={
        boxrule=1.128pt,
        colframe=seyescolA!90!black, colback=seyescolB!10,
        halign=center, valign=center,
        boxsep=2pt, lefttitle=5pt,
        top=4pt, bottom=4pt,
        fonttitle=\bfseries\boldmath
    }
}
```

```
\setrdcrep{correction}
\begin{crep}[seyesfig, width=4cm, adjusted
    title=Figure à tracer :]
\begin{tikzpicture}
    \draw (0,0) circle [radius=1cm] ;
    \draw (-120:1)--(60:1) ;
\end{tikzpicture}
\end{crep}
```

Figure à tracer :



Fractions

`\tcfraccrep`[⟨*options*⟩]{⟨*numérateur*⟩}{⟨*dénominateur*⟩}

Cette commande permet de proposer des fractions à compléter, sensibles au mode correctif.

```
\setrdcrep{correction=false}
Compléter : $\dfrac{24}{36}=
    \tcfraccrep{2}{3}$
\hfill
\setrdcrep{correction=true}
Compléter : $\dfrac{24}{36}=
    \tcfraccrep{2}{3}$
```

Compléter : $\frac{24}{36} = \frac{}{}$ Compléter : $\frac{24}{36} = \frac{2}{3}$

La commande `\tcfraccrep` accepte plusieurs clés de la branche **`/tcb/crep/frac/`**.

`/tcb/crep/frac/width`=⟨*longueur*⟩ (initialement `0.7cm`)

Cette clé permet de modifier la largeur utilisée pour produire les boîtes `\tccrep` qui englobent le numérateur et le dénominateur.

```
\setrdcrep{correction=true}
Compléter : $\dfrac{24}{36}=
    \tcfraccrep[width=2cm]{2}{3}$
```

Compléter : $\frac{24}{36} = \frac{2}{3}$

`/tcb/crep/frac/crep for`=`both|num|den` (par défaut `both`, initialement `both`)

Seul le numérateur ou seul le dénominateur peuvent être sensibles au mode correctif.

```
\setrdcrep{correction=true}
Compléter : $\dfrac{24}{36}=
    \tcfraccrep[crep for=num]{2}{3}$
\hfill
Compléter : $\dfrac{24}{36}=
    \tcfraccrep[crep for=den]{2}{3}$
```

Compléter : $\frac{24}{36} = \frac{2}{3}$ Compléter : $\frac{24}{36} = \frac{2}{3}$

`/tcb/crep/frac/crep num style`=⟨*style /tcb/crep*⟩
`/tcb/crep/frac/crep den style`=⟨*style /tcb/crep*⟩

Ces styles sont passés aux commandes `\tccrep` qui permettent de composer le numérateur et le dénominateur de la fraction.

```
\setrdcrep{correction=true}
Compléter : $\dfrac{24}{36}=
    \tcfraccrep[crep num style={
    colback=yellow}]{2}{3}$
\hfill
Compléter : $\dfrac{24}{36}=
    \tcfraccrep[crep den style={
    colback=yellow}]{2}{3}$
```

Compléter : $\frac{24}{36} = \frac{2}{3}$ Compléter : $\frac{24}{36} = \frac{2}{3}$

`/tcb/crep/frac/crep style`=⟨*style /tcb/crep*⟩

Cette clé est un raccourci pour régler à la fois la clé `/tcb/crep/frac/crep num style` ainsi que la clé `/tcb/crep/frac/crep den style`.

```
\setrdcrep{correction=true}
Compléter : $\dfrac{24}{36}=
    \tcfraccrep[crep style={seyes
    }]{2}{3}$

\tcfillcrep[seyes, normal crep]{%
Compléter : $\dfrac{24}{36}=\
    tcfraccrep[crep style={colback=
    black, opacityback=0.15}]{2}{3}$
}
```

Compléter : $\frac{24}{36} = \frac{2}{3}$

Compléter : $\frac{24}{36} = \frac{2}{3}$

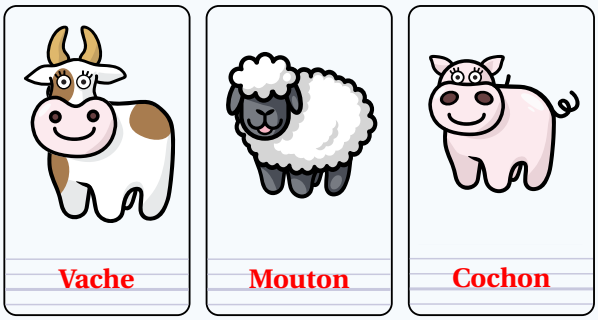
L'environnement seyesrep

```
\begin{seyesrep}[⟨options⟩]{⟨texte⟩}
  ⟨contenu d'environnement⟩
\end{seyesrep}
```

Cet environnement est conçu comme un cas particulier de l'environnement `crep` qui reçoit le style `seyesrep style` défini ci-contre.
Le paramètre optionnel [⟨*options*⟩] permet de fournir des clés supplémentaires tandis que le paramètre obligatoire représente le texte venant s'insérer dans la partie inférieure de la boîte.

```
\tcbset{%
    crep/seyesrep style/.style={
        seyes lower, top=0pt, bottom=1mm,
        middle=0.5mm, boxrule=0.7pt,
        frame style={}, rounded corners, arc=1.5mm,
        halign=center, valign=center, halign lower=center,
        space to upper
    }
}
```

```
\setrdcrep{correction}
\begin{tcbraster}[raster columns=3, raster
    equal height]
    \begin{seyesrep}{Vache}
        \includegraphics[width=2cm]{vache}
    \end{seyesrep}
    \begin{seyesrep}{Mouton}
        \includegraphics[width=2cm]{mouton}
    \end{seyesrep}
    \begin{seyesrep}{Cochon}
        \includegraphics[width=2cm]{cochon}
    \end{seyesrep}
\end{tcbraster}
```



Les clés et l'environnement assosbox

`/tcb/crep/assosbox right`=⟨*longueur*⟩ (par défaut `1.5cm`, initialement `1.5cm`)
`/tcb/crep/assosbox left`=⟨*longueur*⟩ (par défaut `1.5cm`, initialement `1.5cm`)

Ces clés sont destinées à afficher des boîtes permettant des associations (comme, par exemple, associer une proposition à un numéro de réponse) ou pour lesquelles une réponse courte est attendue.
Ces clés installent les styles suivants :

```
\tcbset{%
assosbox common style/.style={
    bicolor jigsaw,
    lower separated,
    frame style={},
    boxrule=0.7pt,
    rounded corners,
    sidebyside align=center
    seam,
    colframe=black
    }
}
```

```
\tcbset{%
    crep/.cd,
    assosbox right/.style={
        crep mode=right,
        assosbox common style,
        righthand width=#1,
        halign lower=center,
        colbacklower=seyescolA
    !15,
        opacityback=0,
    },
    assosbox right/.default=1.5
    cm,
}
```

```
\tcbset{%
    crep/.cd,
    assosbox left/.style={
        crep mode=left,
        assosbox common style,
        lefthand width=#1,
        halign=center,
        colback=seyescolA!15,
        opacityback=1,
        opacitybacklower=0,
    },
    assosbox left/.default=1.5
    cm,
}
```

Le paramètre optionnel désigne la largeur de la boîte inférieure (`/tcb/crep/assosbox right`) ou de la boîte supérieure (`/tcb/crep/assosbox left`).

```
\setrdcrep{correction}
\begin{tcbraster}[raster columns
    =2, raster equal height]
    \begin{crep}[assosbox right]
        Proposition 1
        \tcblower
        C
    \end{crep}
    \begin{crep}[assosbox right]
        Proposition 2
        \tcblower
        A
    \end{crep}
    \begin{crep}[assosbox right]
        Proposition 3
        \tcblower
        D
    \end{crep}
    \begin{crep}[assosbox right]
        Proposition 4
        \tcblower
        B
    \end{crep}
\end{tcbraster}
```

Proposition 1 **C**

Proposition 2 **A**

Proposition 3 **D**

Proposition 4 **B**

```
\setrdcrep{correction}
\begin{tcbraster}[raster columns=2, raster
    equal height]
    \begin{crep}[assosbox left=3cm]
        Oui
        \tcblower
        Question 1
    \end{crep}
    \begin{crep}[assosbox right=3cm]
        Question 2
        \tcblower
        Non
    \end{crep}
\end{tcbraster}
```

Oui Question 1

Question 2 **Non**

Il est permis d'utiliser la clé `/tcb/crep/seyes` avec l'une ou l'autre des clés `/tcb/crep/assosbox right` ou `/tcb/crep/assosbox left`.

```
\setrdcrep{correction}
\begin{tcbraster}[raster columns=1, raster
    equal height]
    \begin{crep}[assosbox right, seyes,
    lefthand ratio=0.5]
        $f(x)=2x+1$
        \tcblower
        affine
    \end{crep}
    \begin{crep}[assosbox right, seyes,
    lefthand ratio=0.5]
        $f(x)=0,5x$
        \tcblower
        linéaire
    \end{crep}
    \begin{crep}[assosbox left, seyes,
    lefthand ratio=0.5]
        $f(x)=-3x+5$
        \tcblower
        affine
    \end{crep}
\end{tcbraster}
```

$f(x) = 2x + 1$ **affine**

$f(x) = 0,5x$ **linéaire**

$f(x) = -3x + 5$ affine

```
\begin{assosbox}[⟨options⟩]{⟨texte⟩}
  ⟨contenu d'environnement⟩
\end{assosbox}
```

Cet environnement est un « raccourci » vers les environnements du type `\begin{crep}[assosbox right]` ou bien `\begin{crep}[assosbox right]`. L'intérêt, ici, réside dans la simplification du code à écrire puisque le paramètre obligatoire désigne la réponse attendue de l'élève. En mode correction, cette réponse est automatiquement placée à gauche ou à droite du cadre en fonction du type de boîte demandé.

```
\setrdcrep{correction}
\begin{tcbraster}[raster columns=1, raster
    equal height]
    \begin{assosbox}[seyes]{Oui}
        En forme ?
    \end{assosbox}
    \begin{assosbox}[box left=2.5cm, seyes]{
    Non}
        Un petit café ?
    \end{assosbox}
\end{tcbraster}
```

En forme? **Oui**

Non Un petit café?

`/tcb/crep/assosbox/box right`=⟨*longueur*⟩ (par défaut `1.5cm`, initialement `1.5cm`)
`/tcb/crep/assosbox/box left`=⟨*longueur*⟩ (par défaut `1.5cm`, initialement `1.5cm`)

Ces clés, utilisables uniquement avec l'environnement `assosbox` permettent de préciser la position du cadre destiné à recevoir la réponse de l'élève : à gauche (`/tcb/crep/assosbox/box left`) ou à droite (`/tcb/crep/assosbox/box right`).
Ces clés peuvent recevoir un paramètre optionnel : la largeur de la boîte réponse (par défaut, 1,5 cm).
Par défaut, c'est la clé `/tcb/crep/assosbox/box right` qui est active lorsque l'environnement `assosbox` est appelé.

Les boîtes \tccheckbox

`\tccheckbox`[⟨*options*⟩]{⟨*contenu*⟩}

Cette commande permet de produire une « boîte à cocher » sensible au mode correctif. Elle est normalement destinée à être utilisée au sein d'un environnement `tcbraster` mais il est permis de s'en servir isolée de tout contexte.

Par défaut, les boîtes du type `\tccheckbox` se voient affecter le style (modifiable par l'utilisateur) défini ci-contre.
Les clés `/tcb/frame code`, `/tcb/left` et `/tcb/before upper pre` sont également utilisées en interne par l'extension pour produire le type de boîte voulu.

```
\tcbset{%
    tccheckbox default style/.style={
            enhanced jigsaw,
            opacityback=0,
            boxsep=3pt, top=0pt,
            right=0pt, bottom=0pt,
            boxrule=0.5pt,
            colframe=black
    }
}
```

```
\setrdcrep{correction}
Êtes-vous d'accord ?
\begin{tcbraster}[raster columns=2,
                  raster equal height]
    \tccheckbox[checked corr]{Oui}
    \tccheckbox{Non}
    \tccheckbox{Peut-être}
    \tccheckbox{NSP}
\end{tcbraster}
```

Êtes-vous d'accord ?

☑ Oui | ☐ Non | ☐ Peut-être | ☐ NSP

```
Une boîte isolée : %
\tccheckbox[checked, on line,  width=4cm,
        tcbox width=forced center]{Toute
    seule}
```

Une boîte isolée : ☑ Toute seule

Au sein d'un environnement `tcbraster`, on utilisera la clé `/tcb/halign` pour modifier l'alignement du ⟨*contenu*⟩.

```
Votre choix ?
\begin{tcbraster}[raster columns=3,
                  raster equal height,
                  halign=center]
    \tccheckbox{Truc}
    \tccheckbox{Machin}
    \tccheckbox[checked corr]{Chose}
\end{tcbraster}
```

Votre choix ?

☐ Truc | ☐ Machin | ☑ Chose

Évidemment, les clés de la familles `/tcb/` sont utilisables pour modifier les boîtes produites par la commande `\tccheckbox` mais d'autres clés spécifiques, de la branche `/tcb/crep/checkbox/`, sont également allouées.

`/tcb/crep/checkbox/left width`=⟨*longueur*⟩ (initialement `1.5em`)

Cette clé permet de régler la largeur de la zone coloriée à gauche de la boîte.

```
Tout va bien ?
\setrdcrep{correction}
\tcbset{crep/checkbox/left width=3em}
\begin{tcbraster}[raster columns=2,
                  raster equal height]
    \tccheckbox[checked corr]{Nickel}
    \tccheckbox{Chrome}
\end{tcbraster}
```

Tout va bien ?

☑ Nickel | ☐ Chrome

/tcb/crep/checkbox/background=⟨*couleur*⟩ (initialement `seyescolA!20`)

Cette clé permet de modifier la couleur de fond utilisée pour colorier la zone à gauche de la boîte.

```
Tout va bien ?
\setrdcrep{correction}
\tcbset{crep/checkbox/background=teal}
\begin{tcbraster}[raster columns=2,
                  raster equal height]
    \tccheckbox{Nickel}
    \tccheckbox[checked corr]{Chrome}
\end{tcbraster}
```

Tout va bien ?

☐ Nickel ☑ Chrome

/tcb/crep/checkbox/check color=⟨*couleur*⟩ (initialement `crepcorrectioncol`)

Par défaut, la couleur de la marque est celle utilisée pour le texte correctif. On peut changer ce réglage à l'aide de la clé `/tcb/crep/checkbox/check color`.

```
Tout va bien ?
\setrdcrep{correction}
\tcbset{crep/checkbox/check color=teal}
\begin{tcbraster}[raster columns=2,
                  raster equal height]
    \tccheckbox{Nickel}
    \tccheckbox[checked corr]{Chrome}
\end{tcbraster}
```

Tout va bien ?

☐ Nickel ☑ Chrome

/tcb/crep/checkbox/checked=`true|false` (par défaut `true`, initialement `false`)

On utilise cette clé avec la valeur `true` pour faire apparaître la marque signifiant que la case est cochée.

```
Tout va bien ?
\begin{tcbraster}[raster columns=2,
                  raster equal height]
    \tccheckbox[checked]{Nickel}
    \tccheckbox[checked]{Chrome}
\end{tcbraster}
```

Tout va bien ?

☑ Nickel ☑ Chrome

/tcb/crep/checkbox/checked corr=`true|false` (par défaut `true`, initialement `false`)

Cette clé, utilisée avec la valeur `true`, montre la marque signifiant que la case est cochée, uniquement lorsque la valeur `true` est affectée à la clé `/crep/correction`.

```
\setrdcrep{correction=false}
\tcbset{mystyle/.style={crep/checkbox/checked corr,
     width=\linewidth, tcbox width=forced center,
    on line}}
Le matin\ldots \begin{tabular}[t]{p{3cm}}
    \tccheckbox[mystyle]{Je m'étire}   \\[5pt]
    \tccheckbox[mystyle]{Je baille}    \\[5pt]
    \tccheckbox[mystyle]{Je me gratte}
\end{tabular}
```

Le matin… ☐ Je m'étire

☐ Je baille

☐ Je me gratte

/tcb/crep/checkbox/check mark=⟨*texte*⟩ (initialement `\ding{52}`)

Cette clé permet de spécifier la marque à utiliser pour signifier que la case est cochée.

```
Tout va bien ?
\begin{tcbraster}[raster columns=2,
                    raster equal height]
    \tccheckbox[checked, check mark=*]
                                {Nickel}
    \tccheckbox[checked,
            check mark=\textbullet]{Chrome}
\end{tcbraster}
```

Tout va bien ?

* Nickel • Chrome

Si l'extension **rdcheckbox.sty** est chargée, il est également possible de se servir de la commande `\rdcrepmark` pour spécifier le type de marque à utiliser.

```
\tcbset{%
crep/checkbox/check mark=\rdcrepmark{mark5}}
Tout va bien ?
\begin{tcbraster}[raster columns=2, raster
    equal height]
    \tccheckbox[checked]{Nickel}
    \tccheckbox[checked]{Chrome}
\end{tcbraster}
```

Tout va bien ?

✗ Nickel ✗ Chrome

`\rdcrepmark`[⟨*options*⟩]{⟨*marque*⟩}

Cette commande est identique à la commande `\checkboxmark` de l'extension **rdcheckbox.sty**. La seule différence réside dans la gestion de la couleur de la marque : celle-ci est automatiquement définie en fonction de la valeur de la clé `/tcb/crep/checkbox/check color`.

`\tccheckboxlist`[⟨*options*⟩]{⟨*liste*⟩}

Cette commande permet d'automatiser quelque peu la création d'une liste de boîtes à cocher.

```
\setrdcrep{correction}
Êtes-vous d'accord ?
Votre couleur préférée ?
\tccheckboxlist[checked corr=3]
                {Bleu, Rouge, Jaune}
```

Votre couleur préférée ?

Bleu Rouge ✔ Jaune

Les clés suivantes, de la branche **/tcb/crep/checkboxlist/**, sont acceptées comme [⟨*options*⟩] de la commande `\tccheckboxlist` :

/tcb/crep/checkboxlist/columns=⟨*nombre entier*⟩ (initialement 0)

Par défaut, la commande `\tccheckboxlist` englobe les éléments de la liste fournie en paramètre obligatoire dans un environnement `tcbraster` dont le nombre de colonnes est égal au nombre d'éléments de la liste.

En saisissant une valeur non nulle pour la clé `/tcb/crep/checkboxlist/columns`, on force l'environnement `tcbraster` à adopter le nombre de colonnes souhaité par l'utilisateur.

```
\setrdcrep{correction}
Votre couleur préférée ?
\tccheckboxlist[columns=2, checked corr=3]
                {Bleu, Rouge, Jaune, Vert}
```

Votre couleur préférée ?

Bleu Rouge

✔ Jaune Vert

/tcb/crep/checkboxlist/col skip=⟨*longueur*⟩ (initialement 2mm)

/tcb/crep/checkboxlist/row skip=⟨*longueur*⟩ (initialement 2mm)

Ces deux clés permettent de modifier l'espacement horizontal et vertical entre les différentes boîtes à cocher.

```
Votre couleur préférée ?
\tccheckboxlist[columns=2, col skip=1cm,
    row skip=1cm]{Bleu, Rouge, Jaune, Vert}
```

Votre couleur préférée ?

☐ Bleu ☐ Rouge

☐ Jaune ☐ Vert

/tcb/crep/checkboxlist/checked=⟨*liste*⟩ (initialement vide)

Cette clé permet de spécifier la ou les boîtes devant être cochées en fournissant, sous forme d'une liste, les numéros des éléments à insérer dans une boîte cochée. Le séparateur de liste est la virgule et, si la liste se résume à un seul élément, il n'est pas nécessaire d'encadrer son numéro par des accolades.

```
Vos couleurs préférées ?
\tccheckboxlist[columns=2, checked={2,3}]
                {Bleu, Rouge, Jaune, Vert}
```

Vos couleurs préférées ?

☐ Bleu ☑ Rouge

☑ Jaune ☐ Vert

/tcb/crep/checkboxlist/checked corr=⟨*liste*⟩ (initialement vide)

Cette clé, sensible au mode correctif, permet de spécifier les boîtes à cocher lorsque la valeur `true` est affectée à la clé `/crep/correction`.

```
\setrdcrep{correction=false}
Vos couleurs préférées ?
\tccheckboxlist[columns=2,
                checked corr={2,3}]
            {Bleu, Rouge, Jaune, Vert}
```

Vos couleurs préférées ?

☐ Bleu ☐ Rouge

☐ Jaune ☐ Vert

/tcb/crep/checkboxlist/raster (style /tcb, initialement vide)

Cette clé permet de passer des options supplémentaires à l'environnement `tcbraster` qui encadre les boîtes à cocher.

```
Vos couleurs préférées ?
\tccheckboxlist[columns=2, checked={2,3},
    raster={halign=center}]
                {Bleu, Rouge, Jaune, Vert}
```

Vos couleurs préférées ?

☐ Bleu ☑ Rouge

☑ Jaune ☐ Vert

/tcb/crep/checkboxlist/checkbox style (style /tcb/crep/checkbox, initialement vide)

Cette clé permet d'affecter un style commun à toutes les boîtes à cocher.

```
Vos couleurs préférées ?
\tccheckboxlist[columns=2, checked=2,
    checkbox style={background=teal}]
                {Bleu, Rouge, Jaune, Vert}
```

Vos couleurs préférées ?

☐ Bleu ☑ Rouge

☐ Jaune ☐ Vert

/tcb/crep/checkboxlist/checked style (style /tcb/crep/checkbox, initialement vide)

Cette clé permet d'affecter un style commun à toutes les boîtes cochées.
Si la clé `/tcb/crep/checkboxlist/checked corr` est utilisée pour définir les cases à cocher, le style donné par la clé `/tcb/crep/checkboxlist/checked style` est appliqué uniquement lorsque la valeur `true` est affectée à la clé `/crep/correction`.

```
Vos couleurs préférées ?
\tccheckboxlist[columns=2, checked={1,4},
    checked style={colframe=red,
        colback=red, opacityback=0.25,
        background=red, check color=white}]
                    {Bleu, Rouge, Jaune,
    Vert}
```

Vos couleurs préférées ?

- [x] Bleu
- [ ] Rouge
- [ ] Jaune
- [x] Vert

/tcb/crep/checkboxlist/checkbox n (style /tcb/crep/checkbox, initialement vide)

Cette clé, qui doit être utilisée comme un style, en remplaçant la lettre *n* par un nombre entier, permet d'affecter un style particulier à une boîte à cocher donnée.

```
Vos couleurs préférées ?
\tccheckboxlist[columns=2,
    checkbox 2/.style={colframe=blue,
    boxrule=1pt}]{Bleu, Rouge, Jaune, Vert}
```

Vos couleurs préférées ?

- [ ] Bleu
- [ ] Rouge
- [ ] Jaune
- [ ] Vert

- Symboles -

- A -

- B -

- C -

- D -

- E -

- F -

- I -

- L -

- N -

- R -

- S -

- T -

- U -

- W -