

18. Fonctionnalités supplétives avec `scrlayer-scrpage`

Le pack `scrlayer-scrpage` fournit des fonctionnalités qui n'ont pas été décrites dans le chapitre 5 de la partie I de ce guide. Celles-ci sont des extensions qui ne sont pas absolument nécessaires pour l'utilisateur moyen et certaines sont mises en œuvre seulement pour la compatibilité avec `scrpage2`, de sorte que la plupart des utilisateurs n'ont pas besoin de les connaître. En conséquences, ces éléments sont documentés ici et seulement dans cette deuxième partie.

Toutefois, pour les nouveaux documents, il est recommandé d'utiliser les fonctions avancées qui ont été décrites dans la section précédente, à la place des commandes de cette section.

18.1. Influence des styles de page définis

Cette section se veut un complément à la section 17.7 et décrit les choses qui ne sont pas immédiatement évidentes pour le débutant.

`\GenericMarkFormat{nom de structure}`

Cette déclaration est utilisée par le formatage par défaut des numéros d'unité dans la gestion des en-têtes des sections et des classes inférieures sans `\chapter` mais également au niveau de section et sous-section, dans la mesure où les marques de déclarations ne sont pas déjà définies ailleurs. La déclaration utilise par défaut `\secCNTmarkformat` si cette instruction interne est définie comme classe de KOMA-Script. `\OtherwisecCNTformat` est avec la déclaration qui est déjà prévue par le noyau de LaTeX pour les classes et forfaits. Comme argument `\GenericMarkFormat` attend le nom de l'organisation, soit par exemple chapitre ou une section précédente SANS barre oblique inverse (engl. `backslash`).

En redéfinissant cette déclaration, le formatage par défaut de tous les numéros d'unité dans le titre des colonnes en cours d'exécution qui en dépendent peut être modifié. De même, une classe à ce sujet peut spécifier un formatage différent sans avoir à changer individuellement toutes les commandes.

Exemple:

Supposons que vous voulez, dans le titre de la colonne, tous les numéros d'articles affichés sous forme de texte blanc dans une boîte noire. Les articles peuvent être définis avec les commandes `scrlayer \sectionmarkformat` et `\subsectionmarkformat` en utilisant `\GenericMarkFormat`, il suffit d'une redéfinition appropriée:

```
\documentclass{article}
\usepackage{blindtext}
\usepackage[automark]{scrlayer-scrpage}
\pagestyle{scrheadings}
\usepackage{xcolor}
\newcommand*{\numberbox}[1]{%
  \colorbox{black}{%
    \strut~\textcolor{white}{#1}~}%
}
\renewcommand*{\GenericMarkFormat}[1]{%
\protect\numberbox{\csname the#1\endcsname}%
\enskip
```

```

}
\begin{document}
\blinddocument
\end{document}

```

Pour les instructions de commutation de la couleur de le pack, xcolor est utilisé. Pour plus de détails, voir les instructions données dans le paquet [Ker07].

En outre, un pilier invisible \strut (entretoise) est inséré. Cette déclaration devrait figurer dans chaque introduction longue de LaTeX.

La boîte avec le nombre définit sa propre commande auxiliaire \NumberBox, elle-même redéfinition à partir de \GenericMarkFormat avec \protect avant l'expansion. Elle est indispensable sinon avec \MakeUppercase pour la bonne réalisation du titre en cours d'exécution, ce ne sont pas les couleurs «noir» et «blanc» mais les couleurs «NOIR» et «BLANC» qui sont nécessaires, mais qui ne sont pas définies.

Alternativement, on peut aussi, en utilisant \DeclareRobustCommand* également redéfinir \NumberBox au lieu de \newcommand* (voir [Tea06]).

Si vous voulez le même livre avec une classe KOMA-Script ou avec les catégories standard, ou encore réaliser un rapport, vous pouvez utiliser \sectionmarkformat et en fonction de la classe \subsectionmarkformat ou \chaptermarkformat puisque ceux-ci en utilisant les classes mentionnées n'utilisent pas \GenericMarkFormat: Voir le chapitre 17.7 pour le détail de cette déclaration

```

\documentclass{scrbook}
\usepackage{blindtext}
\usepackage[automark]{scrlayer-scrpage}
\pagestyle{scrheadings}
\usepackage{xcolor}
\newcommand*{\numberbox}[1]{%
  \colorbox{black}{%
    \strut~\textcolor{white}{\#1}~}%
}
\renewcommand*{\GenericMarkFormat}[1]{%
  \protect\numberbox{\csname the#1\endcsname}%
  \enskip
}
\renewcommand*{\chaptermarkformat}{%
  \GenericMarkFormat{chapter}%
}
\renewcommand*{\sectionmarkformat}{%
  \GenericMarkFormat{section}%
}
\begin{document}
\blinddocument
\end{document}

```

18.2. Définissez votre propre style de page

Dans la section 5.4, les deux styles de page présentés étaient plain.scrheadings et scrheadings. Ils forment quasiment une paire dans laquelle scrheadings fournit le

style du titre principal pendant que plain.scrheadings gère la page de style simple appropriée et sans titre, mais habituellement avec le numéro de page. Dans la suite de ce texte, cet ensemble sera désigné par « paire » ou « couple ».

En plus de la configuration de cette paire, scrlayer-scrpage offre également la possibilité de définir des styles supplémentaires de page avec trois parties séparées dans chaque en-tête et dans chaque pied. La plupart des utilisateurs n'auront pas besoin de plus d'un style prédéfini de paire de page scrheadings. Le nom de la page principale de style scrheadings est presque le nom de la paire de style de page.

Ainsi, les commandes de cette section peuvent être comprises plutôt comme des suppléments pour des cas particuliers, mais je suis presque sûr qu'ils pourraient être également résolus en utilisant seulement la paire prédéfinie.

Et parce que chaque cas est un cas spécial, la section n'a pas non plus d'exemple détaillé d'application pratique. Néanmoins, si je rencontre une belle application, je ne manquerai pas de l'utiliser et de vous la faire connaître dans une future version de la présente.

```
\defpairofpagetypes [paire parent]{nom}{définition}  
\newpairofpagetypes [paire parent]{nom}{définition}  
\renewpairofpagetypes [paire parent]{nom}{définition}  
\providepairofpagetypes [paire parent]{nom}{définition}
```

Avec ces commandes vous pouvez définir des paires de style de page analogues à celles qui peuvent être définies avec scrheadings et plain.scrheadings et dont le nom sera le nom du style de la page principale scrheadings qui est conçu pour une utilisation similaire avec des titres. Le nom correspondant du style de page brute comportera le préfixe plain. Automatiquement. Le nom est donc aussi le nom du couple et de la page principale de style de cette paire, tandis que plain.name est le nom du style de page simple.

Si l'argument facultatif du couple parent n'est pas indiqué, c'est le nom de la paire de style de page avec les paramètres du nouveau couple qui hérite de la configuration de la paire parents et sera initialisé.

La lecture de la section 5.4, peut donner l'impression que les instructions illustrées s'appliquent uniquement aux commandes scrheadings et plain.scrheadings, ce qui est en partie vrai tant qu'elles sont la seule paire de style de page. Dès qu'il existe plusieurs paires de style de page, ces instructions sont liées à la paire activée la dernière. C'est en fait le cas qui s'applique à \lehead, \cehead, \rehead, \lohead, \cohead, \rohead, \lefoot, \cefoot, \refoot, \lofoot, \cofoot, \rofoot, \ihead, \chead, \ohead, \ifoot, \cfoot, \ofoot à la section 5.4.

En plus des dix-huit références indiquées ci-dessus, trois commandes sont à utiliser dans la définition finale des paramètres :

```
\clearmainofpairofpagetypes, \clearpairofpagetypes, \clearplainofpairofpagetypes.
```

Dans ce cas, elles représentent une sorte de configuration de base du style paire de page qui est toujours exécuté lorsque ce style est activé. Un style paire de page est activé par l'un des deux styles paire de page activé. Cela se fait habituellement en utilisant \pagestyle.

Il convient de noter que les instructions de la section 5.5, sont en tous cas de nature générale et s'appliquent à tous les styles de page définis avec scrlayer scrpage.

Alors que `\defpairofpagestyles` définit, quel qu'il soit, le style de page si les styles appropriés existent déjà, utiliser `\providepairofpagestyles` et `\newpairofpagestyles` et ce, uniquement si les mises en page ne sont pas encore définies.

Si au moins un des deux modèles de pages est déjà défini, la définition récente est ignorée dans `\providepairofpagestyles`, et elle conduit, avec `\newpairofpagestyles` à une erreur. Pour la redéfinition des paires existantes `\renewpairofpagestyles` peut être utilisé. Ici, une erreur est signalée si l'un des deux styles de page (ou les deux styles de page du couple) n'existe pas encore.

```
\clearmainofpairofpagestyles
\clearplainofpairofpagestyles
\clearpairofpagestyles
```

Avec `\clearmainofpairofpagestyles`, le principal style de page du dernier couple de style de page chargé, donc actif, est configuré vide. En revanche, l'instruction `\clearplainofpairofpagestyles` permet de configurer la page style plain du couple correspondant pour terminer. Enfin et surtout, les deux styles de page enfin activés comme vide seront configurés par la commande `\clearpairofpagestyles`. Il convient toutefois de noter qu'aucune de ces déclarations ne va supprimer la description de la définition de l'argument spécifié lors de la mise en page de la paire de style de page (voir ci-dessus).

Donc, si vous activez la paire une nouvelle fois, ces définitions seront utilisées à nouveau! Les instructions peuvent aussi être utilisées dans la définition d'un style de page indiquée précédemment. Mais elles peuvent également être utilisées en tout temps en dehors de cette définition. Dans ce cas, elles se réfèrent au dernier style de page actif.

Les trois commandes `\clearscrheadings`, `\clearscrplain` et `\clearscrheadfoot` sont des alias qui assurent seulement la compatibilité avec `scrpage2`. Elles sont obsolètes et ne devraient plus être utilisées.

18.3. Définition des styles de page complexes

En plus des styles prédéfinis et de la détermination de votre propres style, `scrpage2` fournit également la possibilité de mettre en forme des styles de page, où l'en-tête et le pied sont tous deux divisés en trois éléments.

Cette interface existe pour la compatibilité avec `scrpage2`. Néanmoins, elle a été étendue à un ensemble plus cohérent de commandes à la différence de `scrpage2`.

La signification de chaque argument est illustrée dans le schéma d'une double page de la figure 17.1.

Avec ces commandes, vous pouvez définir un seul style de page dont l'en-tête et le pied sont séparés en trois éléments: un élément interne, un élément intermédiaire, central, et un élément externe. Le contenu d'un élément est défini par trois arguments obligatoires pour l'en-tête et trois arguments obligatoires pour le pied. Chacun des deux éléments intérieurs sera imprimé aligné à droite sur la page de gauche et toujours aligné à gauche sur la page de droite. Les éléments centraux seront imprimés centrés respectivement dans l'en-tête et le pied des pages de gauche et de droite. Chacun des deux éléments extérieurs sera imprimé aligné à gauche sur la page de gauche et aligné à droite sur la page de droite.

```

\deftriplepagestyle{nom du style de page}
    [épaisseur ligne extérieure][épaisseur ligne intérieure]
    {tête interne}{tête centrale}{tête externe}
    {pied interne}{pied central}{pied externe}
\newtriplepagestyle{nom du style de page}
    [épaisseur ligne extérieure][épaisseur ligne intérieure]
    {tête interne}{tête centrale}{tête externe}
    {pied interne}{pied central}{pied externe}
\renewtriplepagestyle{nom du style de page}
    [épaisseur ligne extérieure][épaisseur ligne intérieure]
    {tête interne}{tête centrale}{tête externe}
    {pied interne}{pied central}{pied externe}
\providetriplepagestyle{nom du style de page}
    [épaisseur ligne extérieure][épaisseur ligne intérieure]
    {tête interne}{tête centrale}{tête externe}
    {pied interne}{pied central}{pied externe}

```

N'oubliez pas que, dans le mode recto, la seule disposition de côté est la page de droite! Les deux arguments optionnels peuvent être utilisés, dans l'en-tête et le pied du style de page, pour activer les lignes horizontales. Il suffit d'indiquer l'épaisseur désirée de la ligne comme argument.

Le premier argument concerne l'épaisseur des lignes extérieures, à la fois dans l'en-tête et le pied de page. Ces lignes sont tracées sur l'en-tête et sous le pied.

- Dans l'en-tête de page, il sera imprimé au-dessus du contenu de l'en-tête.
- Dans le pied de page il sera imprimé au-dessous du contenu du pied.

Le second argument concerne l'épaisseur des lignes intérieures. Ces deux lignes sont tracées sous l'en-tête et au-dessus du pied.

Si un seul argument est spécifié, ce sera l'épaisseur des lignes intérieures. Dans ce cas, l'épaisseur des lignes extérieures est égale, par défaut, à zéro et aucune ligne extérieure ne sera imprimée. Et si vous omettez les deux arguments optionnels, les lignes sont supprimées.

Noter qu'un argument optionnel vide ne donne pas le même résultat que l'omission de ce même argument! Les arguments optionnels vides ne sont pas autorisés dans ce cas. En revanche, des impressions simples avec l'option largeur de l'en-tête (voir la section 5.5) sont autorisées.

Dans le schéma de la figure 18.1, extérieur, milieu et intérieur sont des exemples de contenus dont les flèches correspondantes illustrent l'expansion de ces contenus. Les textes colorés à l'intérieur des pages indiquent les noms des arguments de ces commandes et les flèches de même couleur correspondent aux éléments de la page. Ici, il n'est pas envisageable, avec les instructions documentées, d'activer des lignes individuelles, mais il est possible d'avoir, par exemple, une ligne en dessous de l'en-tête de page et de n'en avoir aucune dans le pied de page. En cas de besoin, vous devez soit utiliser les commandes de la section précédente ou celles de la section suivante.

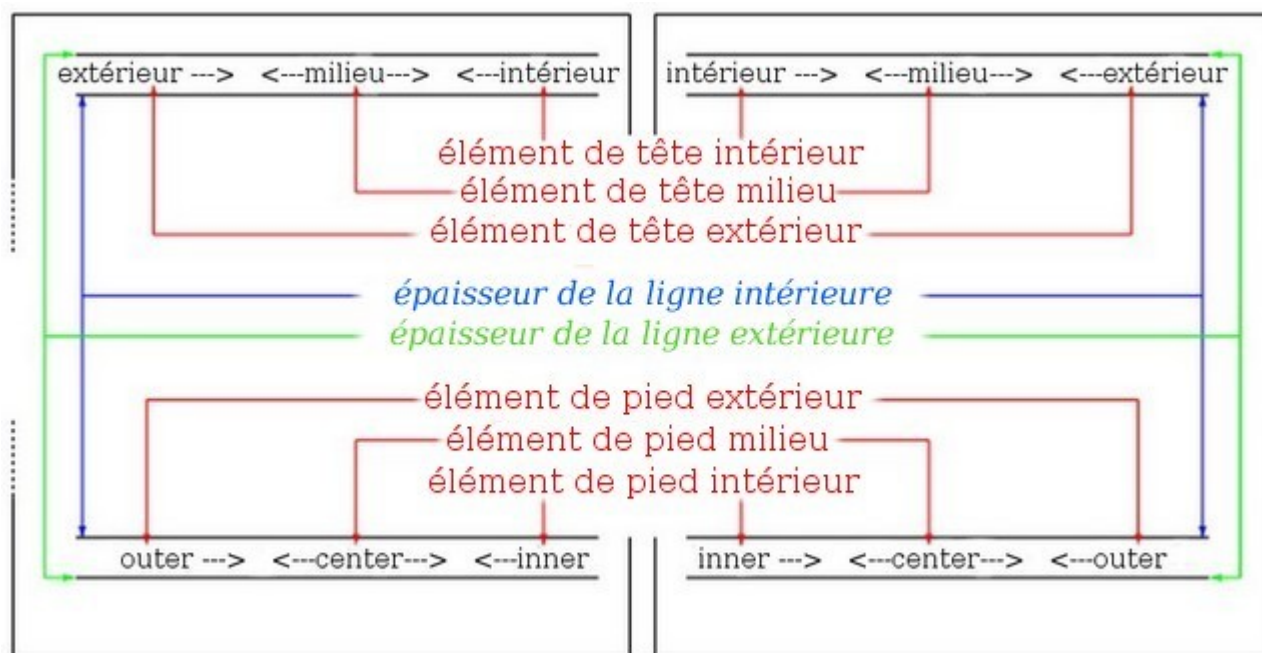


Figure 18.1.: Éléments d'une double page des styles de page tripartites avec les instructions `\deftriplepagestyle`, `\newtriplepagestyle`, `\renewtriplepagestyle` et `\providetripelapagestyle` dans la même couleur que les éléments correspondants de la page.

Attention, la longueur d'un élément unique n'est pas limitée par l'existence d'un élément voisin. Ainsi, dans des circonstances inappropriées, il peut arriver que l'on écrase un élément voisin ou même les deux, et l'utilisateur est seul responsable du résultat dans une pareille situation. Avec têtes de fonctionnement automatiques, cela pourrait, par exemple, être fait en utilisant l'argument optionnel commandes de la section. Consultez le manuel de la classe de document pour plus d'informations sur ces commandes.

La commande `\deftriplepagestyle` définit un style de page indépendamment de savoir si un style avec le même nom existe déjà, alors que seuls, ceux qui n'auraient pas encore été définis le sont avec `\newtriplepagestyle` et `\providetripelapagestyle`. . Toutefois, si un style de page est déjà défini, un message erreur est généré avec `\newtriplepagestyle`. Avec `\providetripelapagestyle`, la nouvelle définition ignore tout simplement l'existant.

Un style de page existant peut être redéfini avec `\renewtriplepagestyle` qui signale une erreur si un style de page avec ce nom n'a pas été précédemment défini.

La commande existante `\deftriplestyle` du pack `scrpage2` correspond à la nouvelle commande `\deftriplepagestyle`. Pour l'unification du nom des instructions, elle a été rebaptisée et ne doit plus être utilisée avec son ancien nom. Néanmoins, l'utiliser se traduirait par un message d'avertissement, mais le résultat serait celui attendu.

18.4. Définition de mises en page complexes

Outre les styles prédéfinis et la création de vos propres nouveaux couples, il est déconseillé d'utiliser des styles obsolètes, alors que le pack `scrpage2` fournit les possibilités les plus fondamentales de définir de nouveaux styles de page. Toutes les définitions de style de page avec `scrpage2` déjà décrites utilisent en

interne ces fonctionnalités de base.

En raison de sa complexité, cette interface est recommandée seulement pour les utilisateurs expérimentés. Les utilisateurs moins avancés devraient être en mesure d'atteindre la quasi-totalité des possibilités précitées en utilisant l'une des interfaces de bases décrites avant. Aussi, ils n'ont pas besoin de lire cette section.

```
\Defpagestyle{name}{définition de l'en-tête}{définition du pied}  
\newpagestyle{name}{définition de l'en-tête}{définition pied}  
\providepagestyle{name}{définition de l'en-tête}{définition du pied}  
\renewpagestyle{name}{définition de l'en-tête}{définition du pied}
```

Ces commandes sont utilisées pour définir un seul style de page avec un maximum de flexibilité. Son nom doit être le nom du style de page qui sera défini. Les arguments de définition de l'en-tête et du pied ont une structure identique,

```
(longueur et épaisseur de la ligne supérieure)%  
{définition de la page de gauche dans la disposition recto-verso}%  
{définition de la page de droite dans la disposition recto-verso}%  
{définition pour toutes les pages mises en page d'un seul côté}%  
(longueur et épaisseur de la ligne inférieure)
```

où les arguments dans les parenthèses sont des options facultatives, ce qui signifie que vous pouvez les omettre ainsi que les crochets. Dans ce cas, l'épaisseur et la longueur de la ligne horizontale correspondante seraient données par les options `headtopline`, `headsepline`, `footsepline` ou `footbotline` de KOMA-Script (voir la section 5.5).

Les trois arguments avec les définition sont obligatoires et utilisés en fonction de la page et des paramètres de mise en page. Le contenu des définitions est arbitraire. Mais il est recommandé d'utiliser une définition interne pour les styles de page dont les en-têtes fonctionnent avec `\headmark`, `\leftmark` ou `\rightmark`. Les numéros ou les titres de section ne doivent être utilisés, en aucun cas, dans la définition. L'ignorer pourrait entraîner des faux numéros ou des erreurs de textes dans l'en-tête ou le pied de page, suite à la création d'une page asynchrone de commandes LaTeX.

La commande `\defpagestyle` va définir le style de page indépendamment de savoir si un style avec le même nom existe déjà ou non, contrairement à la commande `\newpagestyle` qui signale, dans ce cas, une erreur. En revanche la définition de `\providepagestyle` est ignorée si le nom a déjà été utilisé pour un style de page.

Avec `\renewpagestyle` seul un style de page existant peut être redéfini. En interne, cette déclaration est basée sur la commande `\DeclareLayerPageStyle` du pack `scrlayer`.

Les quatre déclarations sont fondées sur `\DeclareLayerPageStyle` du pack `scrlayer`. Les plans définis ici pour un nom du style de page sont présentés au tableau 18.1.

Pour plus d'informations sur les calques et couches des feuilles de style consulter le chapitre 17, dans la partie II.

Tableau 18.1: Les niveaux du style de page définis par `scrlayer-scrpage`

- nom `.head.above.line` définit la ligne horizontale au-dessus de l'en-tête de page,
- nom `.head.odd` définit l'en-tête des pages impaires d'une mise en page recto-verso,
- nom `.head.even` définit l'en-tête de pages paires d'une mise en page recto-verso,
- nom `.head.onside` définit l'en-tête de toutes les pages d'une mise en page d'un seul côté,
- nom `.head.below` définit la ligne horizontale en dessous de l'en-tête.

Puis, pour le pied de la page,

- nom `.foot.above.line` définit la ligne horizontale au-dessus du pied de page,
- nom `.foot.odd` définit le pied des pages impaires d'une mise en page recto-verso,
- nom `.foot.even` définit le pied des pages paires d'une mise en page recto-verso,
- nom `.foot.onside` définit le pied de toutes les pages d'une mise en page d'un seul côté,

et le dernier, mais non le moindre :

- nom `.foot.below` pour la ligne horizontale au-dessous du pied de page.

Exemple:

Supposons que vous voulez le fond de l'en-tête du style de page `scrheadings` avec une seule couleur. Puisque vous avez lu l'introduction de ce chapitre, vous savez que, finalement, un style de page `scrheadings` est composé d'une page de style calque avec, par exemple les couches `scrheadings.head.even`, `scrheadings.head.odd` et `scrheadings.head.onside`.

Dans ce contexte, vous définissez trois couches de fond supplémentaires à ajouter au début du style de page:

```
\documentclass{scrartcl}
\usepackage{automark}{scrlayer-scrpage}
\usepackage{xcolor}
\usepackage{blindtext}
\DeclareLayer[clone=scrheadings.head.onside,
contents={%
    \color{yellow} %
    \rule[-\dp\strutbox]{\layerwidth}{\layerheight}%
}%
]{scrheadings.head.onside.background}
\DeclareLayer[clone=scrheadings.head.odd,
contents={%
    \color{yellow}
    \rule[-\dp\strutbox]{\layerwidth}{\layerheight}%
}
```

```

}%
]{scrheadings.head.odd.background}
\DeclareLayer[clone=scrheadings.head.even,
contents={%
\color{yellow}
\rule[-\dp\strutbox] %
{\layerwidth}{\layerheight}%
}%]
{scrheadings.head.even.background}
\AddLayersAtBeginOfPageStyle{scrheadings}{%
scrheadings.head.onside.background,%
scrheadings.head.odd.background,%
scrheadings.head.even.background%
}
\pagestyle{scrheadings}
\begin{document}
\blinddocument
\end{document}

```

Comme vous pouvez le voir, l'exemple utilise trois couches, de sorte que la position et la taille des couches de fond à venir peuvent être simplement copiées à partir de la couche de tête correspondante en utilisant l'option clone. Ce qui est beaucoup plus facile que d'utiliser une seule couche de fond et d'en calculer dynamiquement la position de temps en temps.

Dans cet exemple, le fond de couleur a été imprimé en utilisant une commande `\rule`, dont les arguments de taille sont donnés par `\layerwidth` et `\layerheight` qui contiennent la largeur actuelle et la hauteur de la couche elle-même.

Un argument optionnel est utilisé pour déplacer le niveau du montant `\dp\strutbox` d'une longueur vers le bas.

Au lieu d'ajouter, comme dans l'exemple, de nouveaux niveaux de la couleur de fond, le problème aurait pu être résolu en utilisant `\colorbox` et `\thead`. Il est recommandé de développer une solution appropriée comme un exercice.

Une autre modification pourrait consister à ajouter des couches d'arrière-plan une par une, juste avant les couches de contenu. Cette manière de faire est également recommandée comme un exercice.

hmode = on-off valeur

Avec le pack `scrpage2`, les en-têtes et les pieds de styles de page sont toujours émis en mode horizontal. Dans `scrlayer-scrpage`, les sorties d'éditions sont elles-mêmes en mode horizontal par défaut. Si vous activez l'option `hmode`, `scrlayer-scrpage` est compatible avec `scrpage2` et dispose librement d'une sortie en mode horizontal.

Mais cela peut avoir des implications, par exemple dans le traitement des espaces blancs au début de la sortie ou dans son alignement vertical.

Cette option peut prendre les valeurs standard pour les interrupteurs simples, comme indiqué dans le tableau 2.5. Elle est désactivée par défaut.