

5. En-têtes et pieds de page avec scrlayer-scrpage

Jusqu'en 2001, il existait la solution du pack scrpage puis, jusqu'à la version 3.11b de KOMA-Script, le pack scrpage2 a été la méthode recommandée pour personnaliser les en-têtes et pieds de page au-delà des possibilités offertes par les styles de page headings, myheadings, plain, et empty des classes standard KOMA-Script ; scrpage est devenu obsolète et en 2013, plus de dix ans plus tard, a finalement été retiré de la distribution régulière de KOMA-Script

En 2013 le pack scrlayer est devenu un module de base de KOMA-Script. Il fournit un niveau de modèle et un système basé sur le style de page. Néanmoins, sa souplesse et la complexité qui en résulte peuvent être trop exigeants à manipuler pour l'utilisateur moyen. Pour plus d'informations sur cette interface, consulter la partie II, chapitre 16.

Beaucoup d'utilisateurs sont déjà familiers avec les instructions de scrpage2 liées à la maîtrise de scrlayer également documenté dans ce chapitre, car nécessaire,, lui aussi, à l'utilisation de scrlayer-scrpage. Ce dernier fournit, en conséquences, une interface utilisateur sur la base de scrlayer grandement améliorée, et largement compatible avec scrpage2. Ainsi, toute personne déjà familière avec scrpage2 et qui n'a pas eu recours à l'utilisation impropre de commandes internes de scrpage2, devrait facilement adopter en remplacement scrlayer-scrpage. La plupart des exemples couvrant scrpage2 dans les livres de LaTeX ou de ressources en ligne devraient également travailler avec scrlayer-scrpage soit directement, soit avec des modifications mineures du code à condition que ces dernières collent aux interfaces standard.

Outre les packs KOMA-Script ci-dessus, vous pouvez en principe utiliser également fancyhdr (voir[vO00]) pour configurer en-têtes et pieds de pages. Cependant, fancyhdr n'a pas de support pour plusieurs options de KOMA-Script, par exemple une mauvaise prise en charge de l'élément police (voir `\setkomafont`, `\addtokomafont`, et `\usekomafont` à la section 3.6) ou le format de numérotation configurable pour les en-têtes dynamiques (voir numéros d'option et `\chaptermarkformat` dans la section 3.16). Par conséquent, si vous usez d'une classe KOMA-Script, l'utilisation du pack scrlayer-scrpage est recommandée. Vous pouvez, bien sûr, employer ce pack avec les classes standard de LaTeX,

Le pack scrlayer-scrpage fournit plusieurs autres fonctions d'un intérêt mineur pour l'utilisateur moyen. Pour cette raison, elles sont développées partie II, à partir du chapitre 17. Consultez les si les options décrites dans la partie I ne suffisent pas à vos besoins.

5.1. Option de sélection précoce ou tardive

Tout ce qui est décrit dans la section 2.4 est généralement applicable mutatis mutandis.

5.2. Hauteur de la tête et du pied

Les classes standard de LaTeX utilisent peu le pied de page dont ils ont inclus le contenu dans un `\mbox`, ce qui se traduit par un pied avec une seule ligne de texte et probablement la raison d'une hauteur de pied mal définie, bien que `\footskip` donne la distance entre la dernière ligne de base de la zone de texte et la ligne de

base du pied de page. Toutefois, si le pied de page se compose de plus d'une ligne de texte, il n'y a pas de déclaration précisant si cette longueur représente la distance de la première ou la dernière ligne de base du pied de page.

Malgré le fait que l'en-tête de page des classes standard soit également mis dans une boîte horizontale considérée comme une seule ligne de texte, LaTeX fournit une longueur d'ajustement de cette hauteur qui est nécessaire pour déterminer le début de la zone de texte.

`\footheight`
`\headheight`

Un pack `\footheight` est introduit par `scrlayer` comme une nouvelle longueur similaire à `\headheight` du noyau LaTeX. En outre `scrlayer-scrpage` utilise `\footskip` qui interprète la distance de la dernière ligne de base possible de la zone de texte à la première ligne du pied de page de base normale.

Le pack `typearea` interprète `footheight` de la même façon, donc, les options de hauteur de pied de `typearea` peuvent également être utilisées pour régler les valeurs de packs `scrlayer` et `scrlayer-scrpage`. (voir les options `footheight` et `footlines` de la section 2.6 ainsi que l'option `footinclude` de la même section).

Si vous n'utilisez pas le pack `typearea`, vous devrez configurer manuellement la hauteur de l'en-tête et du pied en utilisant, si nécessaire, des longueurs chiffrées, et au moins pour la géométrie de l'en-tête, fournir les paramètres correspondants.

Si vous avez installé un en-tête ou un pied de hauteur trop petit pour un contenu efficace, `scrlayer-scrpage` tentera de régler correctement les longueurs correspondantes. En outre, il vous avertira et donnera des informations sur les modifications et les réglages appropriés, que vous pourrez réaliser vous-même. Les changements automatiques sont mis en œuvre aussitôt avec toutefois une réserve : aucune suppression systématique, même si une exigence de hauteur inférieure du contenu est détectée ultérieurement.

5.3. Utilisation des styles de page

Il est applicable mutatis mutandis, ce qui a été écrit dans la section 3.6. Dans ce cas, les noms et les significations des éléments individuels sont listés dans le Tableau 6.1.

Tableau 5.1 : Éléments de `scrlayer-scrpage` dont le type de style peut être changé avec les commandes `\setkomafont` ou `\addtokomafont` de KOMA-Script et paramètres par défaut si ceux-ci ne sont pas définis avant le chargement de `scrlayer-scrpage`

`footbotline`

Ligne en dessous du pied de page d'un style de page défini à l'aide de `scrlayer-scrpage`. La police sera utilisée après `\normalfont` pour les éléments `pageheadfoot` et `pagefoot`. Il est recommandé de n'employer cet élément que pour les changements de couleur. - vide par défaut

`footsepline`

Ligne au-dessus du pied de page d'un style de page défini à l'aide de `scrlayer-`

scrpage. La police sera utilisée après `\normalfont` pour les éléments `pageheadfoot` et `pagefoot`. Il est recommandé de n'employer cet élément que pour les changements de couleur. - vide par défaut

headsepline

Ligne en dessous de l'en-tête d'un style de page défini à l'aide de `scrpage`. La police sera utilisée après `\normalfont` pour les éléments `pageheadfoot` et `pagefoot`. Il est recommandé de n'employer cet élément que pour les changements de couleur. - vide par défaut

headtopline

Ligne au-dessus de l'en-tête d'un style de page défini à l'aide de `scrpage`. La police sera utilisée après `\normalfont` pour les éléments `pageheadfoot` et `pagefoot`. Il est recommandé de n'employer cet élément que pour les changements de couleur. - vide par défaut

pagefoot

Contenu du pied de page d'un style de page défini à l'aide de `scrpage`. La police sera utilisée après `\normalfont` pour l'élément `pageheadfoot`. - vide par défaut

pagehead

Contenu de l'en-tête d'un style de page défini à l'aide de `scrpage`. La police sera utilisée après `\normalfont` pour l'élément `pageheadfoot`. - vide par défaut

pageheadfoot

Contenu de l'en-tête ou du pied d'un style de page défini à l'aide de `scrpage`. La police sera utilisée après `\normalfont`.
- par défaut: `\NormalColor \slshape`

pagenumber

Pagination réglée avec `\pagemark`. Si vous redéfinissez `\pagemark`, vous devez prendre soin que votre redéfinition utilise aussi `\usekomafont{pagenumber}`.
- Par défaut: `\normalfont`

5.4. Utilisation de pages avec styles prédéfinis

La meilleure façon d'obtenir, avec `scrpage`, sa propre conception d'en-tête et de pied de page repose sur l'utilisation d'un style de page prédéfini.

scrheadings

plain scrheadings

Le pack `scrpage` offre deux styles de page à reconfigurer en fonction de vos propres souhaits.

Tout d'abord, le style de page `scrheadings` conçu pour être utilisé avec les titres courants. Ses défauts sont similaires aux rubriques de style de page `headings` de LaTeX ou des classes KOMA-Script. Le contenu de l'en-tête et du pied de page peut être réglé en utilisant les commandes et les options décrites ci-dessous.

Deuxièmement, le style de page `plain scrheadings` prévu pour être utilisé sans en-tête de colonne préconçue. Ses défauts sont similaires au style de page `plain` des classes standard de LaTeX ou des classes KOMA-Script. Les commandes et les options utilisées pour mettre en forme le contenu de l'en-tête et du pied de page

sont décrites ci-après.

Naturellement, vous pouvez configurer `scrheadings` comme style de page en cours d'exécution sans en-tête et `plain.scrheadings` comme style de page avec en-tête. Toutefois, il est conseillé d'adhérer à la convention susmentionnée surtout, parce que les deux styles de page s'influencent mutuellement. Si vous sélectionnez l'un de ces styles de page, `scrheadings` reste accessible en tant que nom de rubrique de style et `plain.scrheadings` sera accessible aussi simplement.

Donc, si vous avez une classe ou un pack à basculer entre rubriques et `plain`, vous devez sélectionner `scrheadings` ou `plain.scrheadings` une seule fois et la classe ou le pack bascule automatiquement sans tenir compte du style de page. Aucun correctif ou adaptation de classes (ou de packs) ne sera nécessaire. Ainsi, ces deux styles de page peuvent être utilisés, sans préalable, en remplacement de titres et de `plain`. De plus amples informations dans l'article 17.2 de la partie II.

Pour les utilisateurs `scrpage2` de l'ancien pack KOMA-Script, il est souligné que `plain.scrheadings` peut être traité pour des raisons de compatibilité avec ce pack sous le nom de l'alias `scrplain`.

```
\lehead[contenu plain.scrheadings]{contenu scrheadings}
\cehead[contenu plain.scrheadings]{contenu scrheadings}
\rehead[contenu plain.scrheadings]{contenu scrheadings}
\lohead[contenu plain.scrheadings]{contenu scrheadings}
\cohead[contenu plain.scrheadings]{contenu scrheadings}
\rohead[contenu plain.scrheadings]{contenu scrheadings}
```

Le contenu de l'en-tête des styles de page `plain.scrheadings` et `scrheadings` peut être paramétré en utilisant ces commandes. Ainsi l'argument optionnel définit le contenu d'un élément du style de page `plain.scrheadings`, tandis que l'argument obligatoire définit le contenu de l'élément correspondant du style de page `scrheadings`.

Le contenu de la page de gauche, donc coté pair que l'on appelle aussi « even », utilise les commandes de réglage `\lehead`, `\cehead` et `\rehead`. Le "e" à la deuxième place du nom de la commande signifie "even" dans le sens de pair.

Le contenu de la page de droite, donc coté impair que l'on appelle aussi « odd », utilise les commandes de réglage `\lohead`, `\cohead` et `\rohead`. Le "o" à la deuxième place du nom de la commande signifie "odd" dans le sens de impair.

Il faut remarquer que seules les pages impaires sont prises en compte par LaTeX dans une simple mise en page pour la numérotation de page.

Chaque en-tête d'un style de page comporte un jeu d'éléments justifiés à gauche avec `\lehead` ou `\lohead`. Remarque: Le "l", pour left, à la première position des noms des commandes signifie "aligné à gauche".

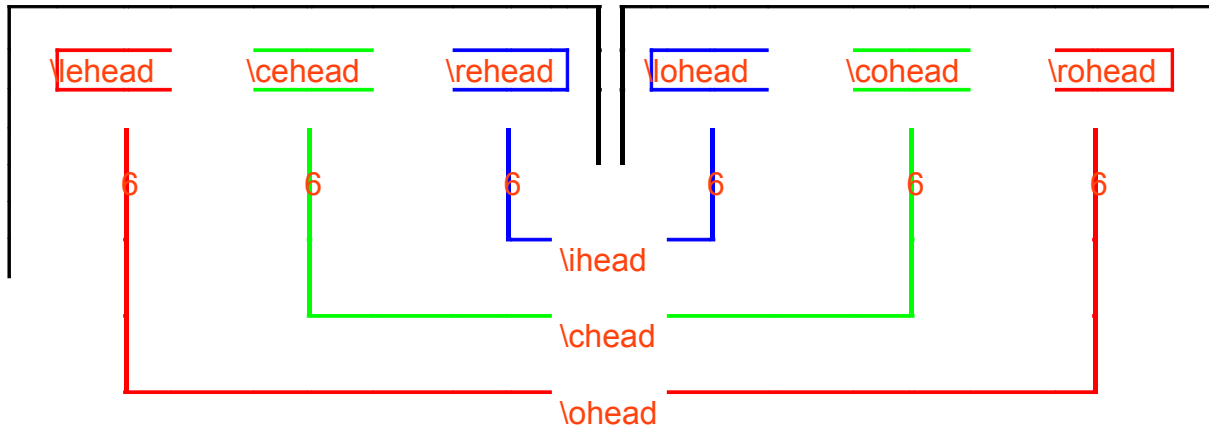
De même, chaque en-tête d'un style de page comporte un jeu d'éléments centrés qui peut être défini avec `\cehead` ou `\cohead`. Le "c", pour centred, au début du nom de la commande signifie "centré".

Enfin, chaque en-tête d'un style de page comporte un jeu d'éléments justifiés à droite avec `\rehead` ou `\rohead`. Le "r", pour right, au début du nom de la commande signifie "aligné à droite".

Ces éléments ne disposent pas de leurs propres polices qui peuvent être modifiées en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir section 3.6), et regroupées toutes ensemble sur l'élément `pagehead`. Avant cet ajout, la fonte de l'élément `pageheadfoot` sera utilisée.

Les paramètres de ces deux éléments sont présentés tableau 6.1 qui montre l'importance des commandes individuelles pour l'en-tête des pages en mode recto-verso et pour la police par défaut de ces éléments.

Figure 5.1 : Sémantique des commandes définissant l'en-tête d'un style de page recto-verso.



Exemple :

Supposons que vous avez écrit un court article et que vous voulez afficher le titre de cet article aligné à gauche et le nom de l'auteur aligné à droite dans l'en-tête de la page. Vous pouvez par exemple utiliser:

```

\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead{Papi Ray}
\rohead{Page style avec \KOMAScript}
\pagestyle{scrheadings}
\begin{document}
\title{Page styles avec \KOMAScript}
\author{Papi Ray}
\maketitle
\end{document}

```

Mais que se passe-t-il ? Sur la première page, seul le numéro de la page apparaît dans le pied, la tête reste vide!

L'explication est très simple. La classe de document scrartcl passe la classe standard sur le style de page plain par défaut.

La commande `\pagestyle{scrheadings}` dans le préambule de notre exemple conduit à l'utilisation réelle du style de page plain.scrheadings pour la page avec en-tête.

L'utilisation d'une classe KOMA-Script par défaut pour ce style de page construit une page avec en-tête vide et un numéro de page dans le pied. Dans l'exemple du code utilisé, les arguments optionnels de `\lohead` et `\rohead` sont omis et le style de page plain.scrheadings reste inchangé par défaut. Le résultat en est donc, pour la première page, effectivement, correct. Puis, ajoutez dans l'exemple de `\maketitle` du texte jusqu'à imprimer une seconde page. Vous pouvez simplement insérer `\usepackage{lipsum}` dans le préambule du document et `\lipsum` après `\maketitle`. Et le haut de la deuxième page contient maintenant, comme vous le souhaitez, le nom de l'auteur et le titre du document.

Pour voir la différence, ajoutez un argument optionnel de `\lohead` et `\rohead` contenant un certain contenu. Pour ce faire, modifier l'exemple ci-dessus comme suit :

```

\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead[Papi Ray]
      {Papi Ray}
\rohead[Page style avec \KOMAScript]
      {Page style avec \KOMAScript}
\pagestyle{scrheadings}
\begin{document}
\title{Page styles avec \KOMAScript}
\author{Papi Ray}
\maketitle
\end{document}

```

Maintenant, vous obtenez également un en-tête sur la première page au-dessus de la tête de titre. Pour ce faire, vous avez simplement reconfiguré le style de page `plain.scrheadings` avec deux arguments optionnels. Mais reconnaissez qu'il serait préférable de laisser le style cette page inchangé, parce que la tête en cours d'exécution au-dessus du titre du document n'est probablement pas le meilleur choix.

Une alternative à la configuration `plain.scrheadings` : utiliser la classe `\titlepagestyle` de KOMA-Script, qui permet aussi de changer les titres d'en-tête. Voir la section 3.12.

Permettez-moi une remarque importante:

Vous ne devriez jamais mettre le titre ou le nombre de niveau de la section directement dans l'en-tête de la page à l'aide d'une nouvelle déclaration par une de ces commandes.

En raison de l'asynchronisme de la mise en page et de la page de sortie de TEX, il peut arriver qu'un mauvais numéro ou un mauvais titre soit affiché dans l'en-tête en cours d'exécution. Au lieu de cela, le mécanisme Mark devrait être utilisé en conjonction avec les automatismes décrits dans la section suivante.

```

\lefoot[contenu plain.scrheadings]{contenu scrheadings}
\cefoot[contenu plain.scrheadings]{contenu scrheadings}
\refoot[contenu plain.scrheadings]{contenu scrheadings}
\lofoot[contenu plain.scrheadings]{contenu scrheadings}
\cofoot[contenu plain.scrheadings]{contenu scrheadings}
\rofoot[contenu plain.scrheadings]{contenu scrheadings}

```

Ce qui est écrit dans le pied de page des styles `plain.scrheadings` et `scrheadings`, est réglable à l'aide de ces commandes. Un argument optionnel définit le contenu de l'élément de `plain.scrheadings`, tandis qu'un argument obligatoire définit le contenu de l'élément correspondant du style de page `scrheadings`. (voir à ce propos le chapitre 6.4 qui traite le problème des en-têtes d'une manière identique)

Comme les commandes d'en-tête, ces éléments ne disposent pas de leurs propres polices qui peuvent être modifiées en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir section 3.6), et regroupées toutes ensemble sur l'élément `pagefoot`.

Avant cet ajout, la fonte de l'élément pageheadfoot sera utilisée. Les paramètres de ces deux éléments sont présentés tableau 6.2 qui montre l'importance des commandes individuelles pour le pied de page en mode recto-verso et pour la police par défaut de ces éléments.

Exemple :

Revenons à l'exemple de l'article court. En supposant que vous souhaitez imprimer l'éditeur sur le côté gauche du pied de page, vous devez modifier l'exemple précédent comme suit :

```
\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead{Papi Ray}
\rohead{Page style avec \KOMAScript}
\lofoot{Editions La Cornée}
\pagestyle{scrheadings}
\usepackage{lipsum}
\begin{document}
\title{Page style avec \KOMAScript}
\author{Papi Ray}
\maketitle
\lipsum
\end{document}
```

Une fois de plus l'éditeur n'est pas imprimé sur la première page avec l'en-tête de titre, en voir l'explication sur `\lohead` dans l'exemple ci-dessus et la solution pour l'imprimer sur la première page serait similaire:

```
\lofoot[Editions La Cornée]{Editions La Cornée}
```

Maintenant, vous décidez de remplacer la police oblique dans l'en-tête et le pied droit par une police droite et plus petite, ce qui peut être réalisé en utilisant:

```
\setkomafont{pageheadfoot}{\small}
```

En outre, l'en-tête doit être en gras, mais pas le pied de page:

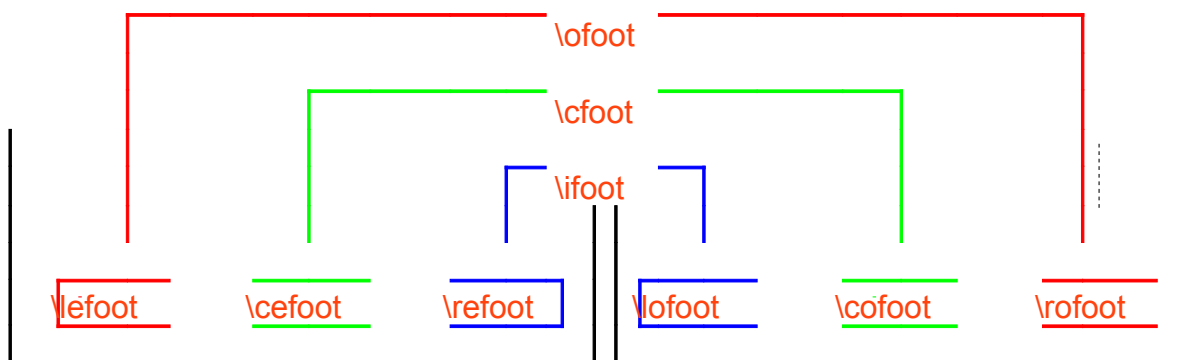
```
\setkomafont{pagehead}{\bfseries}
```

Pour cette dernière commande, il est important de ne la faire qu'après avoir chargé `scrpage-scrlayer` car la classe KOMA-Script définit déjà `pagehead` et `pageheadfoot` avec le même sens.

Le chargement de `scrpage-scrlayer` change le sens de `pagehead` et en fait un élément indépendant de `pageheadfoot`.

Vous pouvez maintenant compléter l'exemple `\lipsum` et même ajouter l'option `twoside` lors du chargement de `scrartcl`. Tout d'abord, le nombre de pages dans le pied se déplace maintenant du centre vers l'extérieur. Cela est dû à la valeur par défaut qui change pour `scrheadings` et `plain.scrheadings` avec une classe de document KOMA-Script recto-verso.

Figure 5.2 : Sémantique des commandes définissant le contenu du pied d'un style de page recto-verso.



Simultanément, l'auteur, le titre du document et l'éditeur disparaissent de la page 2 et apparaissent seulement page 3, ce qui est une conséquence de l'utilisation des commandes que pour les pages impaires. Vous pouvez les reconnaître par le "o" (lofoot, cofoot, rofoot) en deuxième position dans le nom des commandes.

Nous pourrions tout simplement copier ces commandes et remplacer le "o" par un "e" pour définir le contenu des pages paires. Mais avec la mise en page recto-verso, il est plus logique d'utiliser des éléments de miroir inversé. Donc, l'élément de gauche d'une page impaire doit devenir l'élément de droite de la page paire, et vice versa. Pour ce faire, nous remplaçons aussi la première lettre "l" par "r":

```

\documentclass[twoside]{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead{Papi Ray}
\rohead{Page style avec \KOMAScript}
\lofoot[Editions La Cornée]
      {Editions La Cornée}
\rehead{Papi Ray}
\lohead{Page style avec \KOMAScript} \rfoot[Editions La Cornée]
      {Editions La Cornée} \pagestyle{scrheadings}
\usepackage{lipsum} \begin{document}
\title{Page styles avec \KOMAScript} \author{Papi Ray}
\maketitle
\lipsum
\lipsum
\end{document}

```

Après lecture de l'exemple, il peut vous sembler un peu lourd de dupliquer les commandes pour avoir le même contenu d'en-tête ou de pied de page en miroir sur une double page. Il existe une solution facile à mettre en place pour ce scénario tel que présenté ci-dessous.

```

\lohead[contenu plain.scrheadings]{contenu scrheadings}
\chead[contenu plain.scrheadings]{contenu scrheadings}
\rhead[contenu plain.scrheadings]{contenu scrheadings}
\lofoot[contenu plain.scrheadings]{contenu scrheadings}
\cfoot[contenu plain.scrheadings]{contenu scrheadings}
\ifoot[contenu plain.scrheadings]{contenu scrheadings}

```


Pour définir le contenu de l'en-têtes et du pied des pages paires et impaires dans une présentation recto-verso, utilisez les commandes décrites ci-dessus. La teneur de la page paire sera définie à partir de la page impaire. En général, les pages sont symétriques. Un élément aligné à gauche sur une page paire devrait être aligné à droite sur une page impaire et vice versa. Les éléments centrés sur les pages impaires, le sont également sur les pages paires.

Pour simplifier la conception de ces pages, `scrlayer-scrpage` utilise des abréviations. La commande `\ohead` équivaut à appeler `\lehead` et `\rohead`, la commande `\chead` correspond à la fois à `\cehead` et à `\cohead`, et la commande `\ihead` à `\rehead` et à `\lohead`. Cela vaut également pour les instructions du pied. Les relations de tous les en-têtes et pieds de pages sont décrites dans les figures 5.1 et 5.2,.

Exemple : Avec l'aide de nouvelles commandes, le dernier exemple peut être simplifié comme suit:

```
\documentclass[twoside]{scrartcl}
\usepackage{scrlayer-scrpage}
\ihead{Papi Ray}
\ohead{Page style with \KOMAScript}
\ifoot[Editions La Cornée]
{Editions La Cornée}
\pagestyle{scrheadings}
\usepackage{lipsum}
\begin{document}
\title{Page styles avec \KOMAScript}
\author{Papi Ray}
\maketitle
\lipsum
\lipsum
\end{document}
```

Comme vous pouvez le voir, vous pouvez économiser la moitié des commandes et obtenir le même résultat.

Dans les présentations recto, toutes les pages sont les pages impaires. Donc, en mode recto seul, les commandes LaTeX sont synonymes de commandes de pages impaires. Par conséquent, dans la plupart des cas, vous aurez seulement besoin de ces six commandes au lieu des douze décrites précédemment.

`pagestyleset=réglage`

Dans les exemples ci-dessus, vous pouvez trouver quelques informations sur les valeurs par défaut de `scrheadings` et `plain.scrheadings`. En effet `scrlayer-scrpage` fournit deux pré-réglages différents d'origine. Vous pouvez choisir l'une de ces valeurs en utilisant l'option `pagestyleset` manuellement.

Avec le réglage KOMA-Script les valeurs par défaut sont utilisées et seraient également activées automatiquement si une classe KOMA-Script est détectée. Dans ce cas, et dans la mise en page recto-verso, `scrheadings` utilise la gestion de titres extérieurs alignés dans l'en-tête de la page courante. Le numéro de page sera imprimé aligné dans le pied de page. Dans le cas d'un seul côté le titre de la colonne

est centré dans la tête et le numéro de page centré dans le pied. Les majuscules et des minuscules seront automatiquement utilisées pour le titre de l'en-tête, ce qui correspond à l'utilisation de l'option simultanée `markcase=used`. La `pagestyle` de `plain.scrheadings` ne gère pas le paramétrage de l'en-tête, mais les numéros de page qui seront imprimés à l'identique.

Si le réglage est standard, les valeurs utilisées par défaut sont similaires aux rubriques de style de page `headings` et `plain` des classes standard. Ce paramètre sera également activé automatiquement si l'option n'a pas été utilisée et que les classes KOMA-Script ne sont pas détectées. Dans la mise en page recto-verso, `scrheadings` utilise les titres courants alignés en interne dans l'en-tête de page et les numéros imprimés, également dans l'en-tête, alignés à l'extérieur. La mise en page simple face est identique, mais l'en-tête sera alignée à gauche et le numéro de page à droite.

En dépit des importantes considérations typographiques, l'en-tête en cours d'exécution automatique sera convertie en majuscules comme le ferait `markcase=upper`. L'utilisation du style de page `plain.scrheadings` diverge de `scrheading` dans la mesure où le numéro de page est imprimé, en recto-verso, dans le milieu du pied.

Il est important de noter que l'utilisation de cette option est activée en même temps que la page de style `scrheadings`. Ceci vaut également si l'option est utilisée dans un document. Les options de `pack`, `komastyle` et `standardstyle`, documentées pour `scrpage2` n'existent dans `scrlayer-scrpage` que pour des raisons de compatibilité. Elles sont considérées comme obsolètes et ne devraient plus être utilisées.

5.5. Manipulation des styles de page prédéfinis

Dans la section 6.4, nous avons expliqué comment les styles de page `plain.scrheadings` et `scrheadings` sont prédéfinis et comment ces valeurs peuvent être modifiées. Cependant, il manque encore des informations sur, par exemple, la génération d'un titre courant, ou bien comment changer la largeur de la tête et des pieds, ou encore sur la façon de mettre des lignes au-dessus ou en dessous de l'en-tête ou du pied de page. Bien que ces fonctionnalités appartiennent au `pack scrlayer`, elles sont expliquées ci-dessous, les propriétés de base de `scrlayer` sont une partie importante des possibilités de `scrlayer-scrpage`.

```
\Automark[Plan de la section de droite]
      {Plan de la section de gauche}
\AutoMark*[Plan de la section de droite]
      {Plan de la section de gauche}
\manualmark
automark
autooneside=simple interrupteur
manualmark
```

Avec les classes standard de LaTeX et de KOMA-Script, vous pouvez décider d'adopter des en-têtes au fonctionnement automatique ou avec réglage manuel en utilisant un style de page `headings` ou `myheadings` approprié. Comme expliqué dans l'article 3.12, les en-têtes sont obtenus en fonction des titres en cours d'exécution. Les titres courants automatiques sont souvent la réplique d'un extrait de texte important caractéristique de la page, la plupart du temps à l'intérieur de l'en-tête,

parfois, mais plus rarement, dans le pied.

La classe `scrartcl` avec le style de page `headings` utilise le titre de la section en cours ou l'argument optionnel de `\section`, comme en-tête de documents dans le mode simple page. Ceci est considéré comme section par défaut. Les documents recto-verso utilisent cette position intitulée section de gauche tandis que dans le même temps, le sous-titre de section utilise une position de paragraphe à droite. La section de gauche sera imprimée sur les pages impaires, de gauche, d'où l'origine de `left mark`, la section de droite sur les pages paires de droite, et en mode unilatéral, sur toutes les pages. Les classes par défaut éliminent la page paire à chaque fois que l'en-tête de section est dans une page impaire.

Les classes `report` et `book` commencent à un niveau plus élevé. Elles utilisent le titre du chapitre pour la section de droite dans une mise en page à un seul coté. Dans le mode recto-verso, la mise en page utilise le titre du chapitre pour la section de gauche et le titre de la section pour la section de droite.

Si vous utilisez `myheadings` comme style de page, les sections dans l'en-tête existent toujours et seront imprimées, ainsi que les numéros de page même si elles n'ont pas été automatiquement fixées par les rubriques. Ainsi, vous pouvez les remplir manuellement en utilisant uniquement les commandes `\markright`, `\markleft`, et `\markboth`, documentées dans cette section.

Cette différence entre les deux styles de page a été supprimée par `scrpage2` et par `scrlayer`. Au lieu de faire la distinction entre les en-têtes automatique et manuel par le choix d'un style de page, deux nouvelles commandes, `\automark` et `\manualmark`, sont fournies. La commande `\manualmark` passe aux sections manuelles et désactive le paramètre automatique. Contrairement à `\automark` et `\automark*` peut être utilisé pour définir les niveaux de section qui doivent être utilisés pour le réglage automatique. L'argument optionnel indique le repère du niveau de l'organisation à droite, l'argument obligatoire le niveau de la section à gauche.

Les arguments doivent toujours porter le nom d'un niveau de section tel que `partie`, `chapitre`, `section`, `sous-section`, `sous-sous-section`, `paragraphe`, éventuellement `sous-paragraphe` ou `alinéa`.

En général, le niveau le plus élevé doit être utilisé pour la section de gauche et le niveau inférieur pour la section de droite. Il s'agit là d'une simple convention logique et non d'une nécessité.

La différence entre `\automark` et `\automark*` est que `\automark` annule toutes les commandes précédentes pour régler automatiquement les sections, tandis que la version étoilée `\automark*` ne modifie les actions que pour les niveaux de détails spécifiés. Vous pouvez ensuite construire des cas plus complexes.

Exemple :

Supposons que vous voulez le titre du chapitre sur les pages de gauche d'un livre, les en-têtes de chapitre sont utilisés comme titres de fonctionnement des sections sur ce côté au lieu d'être exécutés sur les pages impaires, comme il est habituel pour un livre. Toutefois, pour les pages de droite, vous souhaitez également les en-têtes de chapitre aussi longtemps que le titre du chapitre puisse être utilisé, jusqu'à ce que la première section apparaisse. Pour ce faire, vous devez d'abord charger `scrlayer-scrpage` et sélectionner le style de page `scrheadings`. Ainsi, votre document commence par:

```
\documentclass{scrbook}
\usepackage{scrlayer-scrpage}
\pagestyle{scrheadings}
```

Ensuite, veiller à ce que les en-têtes de chapitre définissent à la fois les sections gauche et droite :

```
\automark[chapter]{chapter}
```

Puis les titres de section doivent également définir les bonnes sections:

```
\automark*[section]{}
```

Voici la version étoilée de l'application utilisée depuis la déclaration précédente `\automark` qui doit encore être valide. En outre, l'argument obligatoire pour le niveau de la section de gauche est laissé vide (empty), parce que cette section doit rester inchangée.

Maintenant, vous avez juste besoin d'un peu de contenu dans votre document pour voir le résultat :

```
\usepackage{lipsum}
\begin{document}
\chapter{chapitre}
\lipsum[1-20]
\section{section}
\lipsum[21-40]
\end{document}
```

Nous utilisons une fois encore le pack Lipsum pour générer un texte bidon avec la commande `\lipsum`. Ce pack est vraiment utile.

Si vous testez l'exemple, vous verrez que la page chapitre de démarrage est sans titre, comme d'habitude, car il ce dernier est automatiquement introduit dans la mise en page du style de page plain de `\plain.scrheadings`. Les pages 2-4 portent les titres d'en-têtes de chapitre : le titre de section page 4, changements du titre de la rubrique page 5. Dès lors, les deux titres sont affichés alternativement dans la tête, sur le côté gauche de l'en-tête de chapitre et sur le côté droit de l'en-tête de la section.

Vous pouvez également utiliser les options `manualmark` et `automark` pour basculer entre les commandes manuelles et automatiques de fonctionnement des en-têtes. Ainsi `AutoMark` utilise toujours par défaut

```
\automark[section]{chapter}
pour les classes avec \chapter et pour les classes sans \chapter
\automark[subsection]{section}
```

Si, en mode unilatéral, vous ne voulez pas que le niveau inférieur influe sur la section de droite mais que, par exemple, seul le niveau supérieur en mode recto-verso affecte la section de gauche et indique les titres des en-têtes de toutes les pages, l'option par défaut `autooneside` permet ce comportement. Elle comprend les valeurs pour les interrupteurs simples, qui peuvent être trouvées dans le tableau 2.5. Vous pouvez la désactiver pour une option de mise en page simple face, les arguments obligatoires de `\automark` et `\automark *` influenceront encore l'en-tête en cours d'exécution.

Exemple :

Supposons que vous voulez un rapport simple face avec un en-tête similaire à celui du livre dans l'exemple précédent. Concrètement, le titre du chapitre devrait être utilisé aussi longtemps que la première section apparaît. A cet effet, l'exemple est modifiée comme suit:

```
\documentclass{scrreprt}
\usepackage[autooneside=false]{sclayer-scrpage}
\pagestyle{scrheadings}
\automark[section]{chapitre}
\usepackage{lipsum}
\begin{document}
\chapter{Titre du chapitre}
\lipsum[1-20]
\section{Titre de la section}
\lipsum[21-40]
\end{document}
```

Vous remarquerez que, dans ce cas, nous n'utilisons pas la commande `\automark*`. Vous pouvez aussi essayer l'exemple avec `[autooneside=true]` ou supprimer l'option et sa valeur. Vous devriez trouver une différence dans l'en-tête de la page 4 en cours d'exécution.

Notez que le chargement du pack n'a pas d'effet sur le fait de savoir si vous utilisez un en-tête automatique ou manuel, ou si ce genre de titre ne remplit pas les sections. Seule l'utilisation explicite des options `automark` ou `manualmark`, ou des commandes `\automark` ou `\manualmark` produit ici une situation claire.

`markcase=valeur`

Comme déjà mentionné plus haut, vous pouvez, avec `sclayer`, basculer le fonctionnement des en-têtes entre manuel et automatique. Dans les titres courants, la section équivalente sera avisée des commandes de sectionnement. Certaines cultures convertissent les titres courants en majuscules contrairement à l'habitude typographique allemande. Les classes standard de LaTeX le font par défaut ainsi que le pack `sclayer` fourni en option.

Pour contrôler cette fonction, vous devez utiliser l'option `markcase=upper` qui se traduit par une redéfinition de la commande `\MakeMarkcase` utilisée par `sclayer` pour le fonctionnement automatique des en-têtes. Il faut reconnaître que les résultats de la commande de LaTeX, `\MakeUppercase`, qui propose la composition avec des lettres supérieures qui n'utilisent pas, ni l'espacement, ni l'équilibrage de ces lettres ne sont pas satisfaisants. Une des raisons de ce comportement pourrait mener à la nécessité d'une analyse de glyphe pour intégrer la forme concrète des lettres et de leur combinaison dans l'équilibre du bloc.

Par conséquent, l'auteur de KOMA-Script préconise de renoncer à l'usage de cette commande pour l'exécution typographique d'en-têtes. Ceci pourrait être réalisé en utilisant `markcase=used`. Néanmoins, certaines classes ajoutent `\MarkUppercase` ou même la commande TEX `\uppercase` dans les titres courants.

Pour de tels cas, l'option `markcase=noupper` peut être utilisée. Cela permettra également de désactiver `\MakeUppercase` et `\uppercase` dans l'en-tête local.

Vous pouvez trouver toutes les valeurs valides pour `markcase` dans le tableau 6.2.

Tableau 5.2 :

Valeurs possibles de l'option markcase pour le choix des minuscules/majuscules dans la composition des en-têtes de colonnes automatiques.

lower

redéfinit `\MakeMarkcase` pour convertir les titres courants automatiques en lettres minuscules en utilisant `\MakeLowercase` (minuscules composition).

upper

redéfinit `\MakeMarkcase` pour convertir les titres courants automatiques en lettres majuscules en utilisant `\MakeUppercase` (majuscules composition).

used

redéfinit `\MakeMarkcase` pour utiliser des en-têtes de fonctionnement automatique, sans aucune modification de cas.

ignoreuppercase, nouppercase, ignoreupper, noupper

redéfinit non seulement `\MakeMarkcase` mais empêche, pour le titre courant, d'effectuer un changement automatique minuscule/majuscule. Localement désactivé pour tous les niveaux de tous les styles de page des en-têtes exécutant `\MakeUppercase`.

`\leftmark`

`\rightmark`

`\headmark`

`\pagemark`

Si vous voulez écarter de la page des styles prédéfinis, habituellement, vous devez décider où placer le contenu des sections. Avec `\leftmark` vous pouvez indiquer le contenu de la section de gauche, comme vous pouvez utiliser `\rightmark` pour indiquer le contenu de la section de droite. Pour plus d'informations sur certaines subtilités de cette commande, voir la description de `\rightmark` à l'article 20.1.

D'une certaine manière l'utilisation de l'extension d'alias de `scrlayer` `\headmark`, selon que la page en cours est paire, `\rightmark` ou impaire, `\leftmark`, serait plus facile.

La commande `\pagemark` n'a strictement rien à voir avec les sections de TEX. Elle est utilisée pour placer un numéro de page formaté. L'élément de police `\pagenumber` est utilisé pour la sortie. Ceci peut être changé en utilisant les commandes `\setkomafont` ou `\addtokomafont` (voir également la section 3.6).

Exemple :

Supposons que vous souhaitez utiliser un en-tête aligné à gauche et aligner le numéro de page dans l'en-tête des pages à droite d'un document unique avec une mise en page à un seul côté. Les exemples suivants minime, mais efficaces le feraient:

```
\documentclass{scrreprt}
\usepackage{blindtext}
\usepackage[automark]{scrlayer-scrpage}
\pagestyle{scrheadings}
\ihead{\headmark}
```

```
\ohead[\pagemark]{\pagemark}
\chead{}
\cfoot[]{}
\begin{document}
\blinddocument
\end{document}
```

Voici le pack Blindtext avec sa commande `\blinddocument` utilisé pour générer facilement du contenu pour un exemple de document.

Les commandes `\ihead` et `\ohead` ont été utilisées pour placer les sections recherchées. Le marqueur du numéro de page de l'argument optionnel `\ohead` est configuré non seulement avec `scrheadings pages`, mais aussi pour être utilisés automatiquement sur la première page du style simple de `plain.scrheadings`. Depuis que les deux styles de page sont prédéfinis avec des sections dans le centre de l'en-tête et du pied, ces éléments sont autorisés en `\chead` et `\cfoot` avec des arguments vides. Sinon, vous pouvez également utiliser `\clearpaïrofpagetypes` avant `\ihead`. Vous trouverez une description de cette commande à la section 17.1.

Notez que l'argument optionnel vide dans le `\cfoot` de l'exemple n'est pas synonyme de l'omission de cet argument qui est facultatif. Faites vous-même un essai et de regarder le bas de la première page.

```
\partmarkformat
\chaptermarkformat
\sectionmarkformat
\subsectionmarkformat
\subsubsectionmarkformat
\paragraphmarkformat
\subparagraphmarkformat
```

Habituellement, les classes KOMA-Script et le pack `sclayer` utilisent ces commandes en interne pour indiquer les numéros de section en cours d'exécution automatique, avec en plus la prise en charge du mécanisme `\autodot`. Si vous le souhaitez ces commandes peuvent être redéfinies pour obtenir une autre forme de numéros de section.

Exemple :

Supposons que vous vouliez que les titres de section soient définis dans l'en-tête sans numéro de classement, ce qui est facile à réaliser avec:

```
\renewcommand*{\sectionmarkformat}{}
```

```
\partmark{texte}
\chaptermark{texte}
\sectionmark{texte}
\subsectionmark{texte}
\subsubsectionmark{texte}
\paragraphmark{texte}
\subparagraphmark{texte}
```

Ces instructions sont utilisées en interne par la plupart des classes pour définir les sections en parlant des commandes de sectionnement. Dans ce cas, seul le texte sans le numéro est attendu comme un argument. Ce nombre est déterminé

automatiquement à la place du numéro actuel, si vous travaillez avec des titres numérotés. Si ces instructions sont redéfinies par l'utilisateur, il doit, avant de fixer le nouveau nombre, impérativement vérifier sur `secnumdepth` les chiffres du niveau de la section, même dans le cas où il ne change pas lui-même `secnumdepth`, parce que les packs et les classes peuvent le faire localement et compter sur une manipulation correcte de `secnumdepth`!

Le pack `scrlayer` définit également ces instructions partiellement renouvelées après chaque appel à `\manualmark` ou `\automark` ou les options appropriées pour désactiver ou activer le titre de fonctionnement automatique ou manuel souhaité.

```
\markleft{section gauche}  
\markright{section droite}  
\markboth{section gauche}{section droite}
```

Indépendamment du fait que les titres soient manuels ou automatiques, vous pouvez modifier le contenu des sections gauche ou droite avec l'une de ces instructions à tout moment. Il est à noter que le repère de gauche généré dans `\leftmark` est la section de gauche de la dernière commande `\markboth` ou `\markleft` de la page en cours, tandis que le repère de droite généré dans `\rightmark` est la section droite de la première commande `\rightmark` ou `\markboth` de la page en cours. Plus de détails sur `\rightmark` section 20.1.

Lorsque vous travaillez avec des titres manuels, les sections restent valables jusqu'à ce qu'elles soient explicitement remplacées par une nouvelle aide de l'instruction équivalente. Lors de l'exécution automatiques de l'en-tête, les repères peuvent devenir invalides avec la mutation de la tête de section en fonction de la configuration de l'automatisme.

Vous pouvez également utiliser ces manoeuvres avec les versions étoilées des commandes de la section.

Exemple :

Supposons que vous écrivez une préface de plusieurs pages juste avant la table des matières qui ne doit pas apparaître. Les titres des colonnes doivent mettre la préface en évidence et vous voulez aussi un en-tête en cours d'exécution pour cette préface:

```
\documentclass[headsepline]{book}  
\usepackage{scrlayer-scrpage}  
\pagestyle{scrheadings}  
\usepackage{blindtext}  
\begin{document}  
\chapter*{Preface}  
\markboth{Preface}{Preface}  
\blindtext[20]  
\tableofcontents  
\blinddocument  
\end{document}
```

Tout d'abord cela semble produire le résultat voulu. Mais un examen plus attentif permet de noter que la colonne intitulée «Préface» n'est pas, contrairement aux autres titres de colonnes, en lettres majuscules.

Vous pouvez facilement résoudre ce problème:


```

\documentclass[headsepline]{book}
\usepackage{sclayer-scrpage}
\pagestyle{scrheadings}
\usepackage{blindtext}
\begin{document}
\chapter*{Preface}
\markboth{\MakeMarkcase{Preface}}
        {\MakeMarkcase{Preface}}
\blindtext[20]
\tableofcontents
\blinddocument
\end{document}

```

Comme on peut le voir, la commande `\MakeMarkcase` a été utilisée pour ajuster également le titre manuellement corrigé et mettre la préface en conformité avec les titres automatiques du reste du document.

Maintenant, passons la commande `\tableofcontents` en face de la préface. Enlevons aussi la commande `\markboth`. Vous constatez que la préface porte maintenant comme titre courant "contents" en cours d'exécution. Ceci est dû à une particularité de `\chapter*` (voir section 3.16). Si, en pareil cas, la tête en cours d'exécution doit être retirée, `\markboth` serait utile avec deux arguments vides:

```

\documentclass[headsepline]{book}
\usepackage{sclayer-scrpage}
\pagestyle{scrheadings}
\usepackage{blindtext}
\begin{document}
\tableofcontents
\chapter*{Preface}
\markboth{}{}
\blindtext[20]
\blinddocument
\end{document}

```

headwidth=largeur: offset
footwidth=largeur: offset

Par défaut, l'en-tête et le pied de la page ont la même largeur que la zone de texte. Cela peut être modifié en utilisant ces options KOMA-Script. La valeur de la largeur indiquée est la largeur voulue respectivement pour l'en-tête et le pied de page. Le décalage définit de combien la tête ou le pied doivent être déplacés vers la marge extérieure, dans une mise en page simple, le côté droit. Les deux valeurs sont facultatives et peuvent être omises. Dans ce cas, vous pouvez également omettre le double point.

Une seule valeur sans les deux points indique la largeur.

Pour la largeur, ainsi que le décalage, vous pouvez utiliser une valeur de longueur valide, longueur de LaTeX, de TEX ou TEX distance. En outre, vous pouvez utiliser une expression de dimension ε -TEX avec des opérations arithmétiques de base +, -, *, / et les parenthèses. Voir [Tea98, section 3.5] pour plus d'informations sur ces expressions et la section 6.1 pour plus d'informations sur l'utilisation d'une longueur de LaTeX comme option de valeur.

La largeur peut alternativement être l'une des valeurs symboliques indiquées dans le tableau 6.3.

La valeur par défaut pour la largeur correspond à la largeur de la zone de texte. La valeur par défaut pour le décalage dépend de la largeur choisie. La moitié de la différence entre la largeur et la largeur de la zone de texte est, en général, utilisée dans un sens unique. Ainsi, la tête au-dessus du champ de texte est centrée.

Dans une mise en page recto-verso, seul un tiers de la différence entre cette largeur et la largeur de la zone de texte est utilisée. La largeur est donc la largeur de l'ensemble de la zone de texte augmentée de la colonne de note marginale, et le décalage sera de zéro par défaut.

Pour éviter des calculs trop compliqués, il suffit simplement de spécifier le décalage désiré.

headtopline=épaisseur: longueur
headsepline=épaisseur: longueur
footsepline=épaisseur: longueur
footbotline=épaisseur: longueur

Les classes KOMA-Script ne fournissent que deux lignes de séparation activables et désactivables uniquement en option, l'une en dessous de l'en-tête et l'autre au-dessus du pied de page.

Le pack scrlayer-scrpage fournit quatre lignes horizontales: une au-dessus et une en dessous de l'en-tête, une au-dessus et une en dessous du pied. Chaque ligne peut être activée et configurée séparément en longueur et épaisseur.

Les deux valeurs sont en option.

Si vous omettez l'épaisseur, une valeur par défaut de 0,4 pt sera utilisée, dite un délié. Si vous omettez la longueur, la largeur de la tête respective sera utilisée pour le pied. Si vous omettez les deux, vous pouvez également ne pas mettre le double point. Si vous utilisez une seule valeur sans les deux points, ce sera l'épaisseur.

Bien entendu, la longueur doit être non seulement inférieure à la largeur courante de la tête, mais aussi du pied. Pourtant, elle peut également être plus longue. Voir les options llines, clines et olines, expliquées plus loin dans cette section. En plus de l'épaisseur et de la longueur la couleur des lignes peut aussi être modifiée, ce qui, tout d'abord, dépend bien sûr de la couleur définie pour la tête et le pied.

Tableau 5.3 : Valeurs symboliques autorisées pour les options de largeur d'en-tête et de pied de page

foot
la largeur actuelle du pied de page

footbotline
la longueur actuelle de la ligne horizontale au-dessous du pied de page

footsepline
la longueur actuelle de la ligne horizontale au-dessus du pied de page

head

la largeur actuelle de l'en-tête de la page

headsepline

la longueur actuelle de la ligne horizontale au-dessous de l'en-tête de la page

headtopline

la longueur actuelle de la ligne horizontale au-dessus de l'en-tête de la page

marginpar

la largeur actuelle de la colonne de note marginale y compris la distance entre la zone de texte et la colonne de la note dans la marge

page

la largeur de la page, en tenant compte d'une possible correction de la liaison définie en utilisant le pack typearea (en option BCOR voir section 2.6)

paper

la largeur actuelle du papier sans tenir compte d'une correction de liaison

text

la largeur actuelle de la zone de texte

textwithmarginpar

la largeur actuelle de la zone de texte et de la colonne marginale y compris la distance entre les deux (Attention: dans ce cas et seulement dans ce cas, la valeur par défaut pour compenser est zéro)

Mais indépendamment des réglages par défaut des éléments, headtopline, headsepline, footsepline et footbotline seront utilisés en supplément pour les correspondants. Vous pouvez les modifier ces utilisant les commandes `\setkomafont` ou `\addtokomafont` (voir section 3.6). Par défaut ces paramètres sont vides, ce qui signifie aucun changement de la police courante ni de la couleur qui ne seraient pas logiques et ne sont pas recommandés pour ces éléments.

Le pack `scrpage2` comporte en outre des options correspondant aux quatre commandes principales `\setheadtopline`, `\setheadsepline`, `\setfootsepline` et `\setfootbotline`. Celles-ci ont un premier argument optionnel pour la longueur, un second argument obligatoire pour l'épaisseur et un troisième argument optionnel pour le réglage de police ou la couleur. Ces instructions existent dans Le pack `scrlayer-scrpage` pour des raisons de compatibilité. Elles sont néanmoins considérées comme obsolètes et ne devraient plus être utilisées. Pour être clair: ces commandes n'avaient jamais été faites pour changer ou supprimer les lignes. Au contraire, elles étaient les seules à permettre d'ajuster la longueur et l'épaisseur de ces dernières, ce qui a été malheureusement souvent ignoré par les utilisateurs!

`plainheadtopline`=simple interrupteur

`plainheadsepline`=simple interrupteur

`plainfootsepline`=simple interrupteur

`plainfootbotline`=simple interrupteur

Avec ces options, les paramètres pour les lignes peuvent également être appliqués pour le style simple de la page. La valeur marche-arrêt est la valeur standard disponible pour les interrupteurs simples, comme indiqué dans le tableau 2.5. Si cette option est activée, les paramètres sont appliqués. Lorsqu'elle est désactivée, le style simple page ne sera pas affiché sur la ligne correspondante.

ilines
clines
olines

Nous avons déjà vu qu'une ligne horizontale au-dessus ou en dessous de l'en-tête ou du pied de page peut être plus longue ou plus courte que cet en-tête ou ce pied de page. Seule la réponse à la question sur l'alignement de cette ligne est toujours manquante. Par défaut, les lignes sont alignées du côté gauche dans la mise en page simple face et justifiées sur la marge intérieure de l'en-tête ou du pied dans la mise en page recto-verso. Ceci correspond à l'option ilines. Alternativement, vous pouvez utiliser l'option clines pour centrer les lignes de la tête ou du pied et l'option olines pour les aligner à droite sur la marge extérieure.