Rechercher et remplacer à l'aide de «RegEx» (2)

Hans Häsler, Lausanne

«Lookbehind» et «Lookahead», tels étaient les mots-clés utilisés dans le numéro 4.2007 du *Bulletin technique* pour annoncer la suite de l'article relatant le rechercher et remplacer étendu d'InDesign CS3. Cela rend accessible ces actions à chaque utilisateur. Plus besoin de recourir à des scripts comme dans les versions antérieures.

Ces deux expressions peuvent être traduites littéralement. *Lookbehind* veut dire regarder derrière et *Lookahead* signifie regarder devant. En évoquant cette technique d'une manière générale, on utilise le terme *Lookaround* (regarder autour).

A quoi servent-ils?

A l'instar du métacaractère «début de ligne» mentionné dans l'article précédent, ils sont employés comme ancres. Une différence importante: les *Lookarounds* ne retournent pas les signes trouvés, mais «oui» ou «non».

Dans InDesign on peut les utiliser quand il s'agit de ne remplacer qu'une partie du motif trouvé. Ou lorsqu'ils doivent servir comme ancre pour insérer des caractères.

Ouvrir le dialogue «Rechercher/remplacer par...» et sélectionner l'onglet «GREP». Le bouton derrière le champ texte «Rechercher» contient le menu local «Caractères spéciaux pour la recherche».

Lookbehind et *Lookaround* (fig. 1) permettent la composition des motifs RegEx qui, sans eux, seraient très compliqués ou bien pas réalisables du tout.

Comment les motifs se présentent-ils?

On les reconnaît au point d'interrogation se trouvant derrière une parenthèse ouvrante.

(?<=x)y	= Lookbehind positif
(?<!x)y	= Lookbehind négatif
y(?=x)	= Lookahead positif
y(?!x)	= Lookahead négatif

Quand le point d'interrogation est suivi d'un signe égal ou d'un point d'exclamation, il s'agit d'un *Lookahead* qui, lui, regarde en avant, dans le sens de la lecture.

Le signe «<» (= plus petit que) indique la direction dans laquelle un *Lookbehind* regarde en arrière. Le signe égal signifie une correspondance positive, le point d'exclamation est utilisé quand le caractère suivant (ou précédent) ne doit pas être présent.

Les lettres en rouge marquent l'emplacement de signes littéraux ou d'une RegEx.

Quelques exemples très simples

Les motifs ci-après sont simplifiés. Ainsi, il est plus facile de reconnaître quelle chaîne doit être placée à quelle position.

Le Lookahead

Un *Lookahead* négatif est indispensable lorsqu'il s'agit de trouver quelque chose qui n'est pas suivi d'un certain signe ou d'une chaîne de caractères spécifique. Dans ce cas, une négation traditionnelle (par le signe caret ^) ne peut pas être utilisée. Mais un *Lookahead* négatif offre la solution: **a(?!b)**.

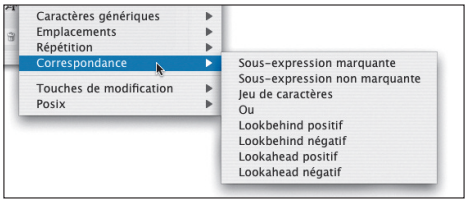

Fig. 1 – Les quatre derniers sous-articles de l'article «Correspondance» (menu pop-up «Caractères pour la recherche) simplifient beaucoup les motifs RegEx.

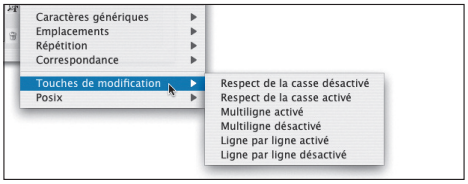

Fig. 2 – Les noms des sous-articles de l'article «Touches de modification» sont autoexpliants.

Les caractères «a» et «b» sont insérés littéralement. En ajoutant un métacaractère pour le début d'un mot, le motif **\<a(?!b)** permet de trouver tous les mots commençant par un «a» mais dont la deuxième lettre n'est pas un «b».

Il suffit de remplacer le point d'exclamation par un signe égal et le motif est transformé en un *Lookahead* positif: **\<a(?=b)**. Celui-ci permet de trouver des mots qui commencent par «ab».

Lors du clic sur le bouton «Modifier», le «b» est épargné. On ne peut qu'atteindre le «a». Il y a trois possibilités:

c$0 = un «c» est inséré devant le «a»;
c = le «a» est remplacé par un «c»;
$0c = un «c» est inséré derrière le «a».

Le Lookbehind

Celui-ci indique à la machine RegEx de vérifier lors d'une correspondance trouvée, si oui (= positif = présent) ou non (= négatif = absent) cette dernière est précédée du motif indiqué.

Un *Lookbehind* fonctionne comme un *Lookahead*, mais dans l'autre sens. **(?<!a)b** trouve un «b» qui n'est pas précédé directement par un «a». Le «b» dans «abc» sera ignoré, mais pas celui dans «acb».

Le Lookbehind est limité

On peut mettre des motifs RegEx à l'intérieur des parenthèses. Mais contrairement au *Lookahead,* il n'est pas possible d'utiliser des positions non définies.

Un exemple. Lorsque le bloc texte contient «un tessst», la recherche à l'aide du motif **(?<=s+)t** ne trouve pas de correspondance. Mais avec **(?<=s{3})t** le «t» final est sélectionné. Dans le premier cas, le nombre des «s» n'est pas défini. Dans le second, il y a une valeur précise: **{3}**. Cela correspond au nombre présent, la lettre est trouvée.

Le *Lookahead*, en revanche, trouve le «e» sans problème avec **e(?=s+)**. La recherche avance dans le sens de la lecture, le motif ne doit pas absolument consister de signes à un nombre fixe.

Un grand avantage

Comme déjà mentionné, les *Lookarounds* sont spécialement utiles quand il s'agit de ne modifier qu'une partie du motif. Certes, on pourrait obtenir un résultat identique en utilisant des groupes. Mais cela n'est valable que pour une recherche positive. Pour une assertion négative on doit avoir recours à ce genre de RegEx pas facile à maîtriser.

Les touches de modification

Un nom d'article trompeur (fig. 2). Ce ne sont pas des touches, mais des bouts de codes insérés lors du choix de l'un des sous-articles. Ils servent à modifier l'action du motif placé derrière.

Respect de la casse désactivé

Le code **(?i)** inséré ne fait pas de différence entre lettres majuscules et lettres minuscules. Le motif **(?i)test** sélectionne à tour de rôle les trois mots «Test test TEST».

Respect de la casse activé

La seule différence: le trait d'union inséré, qui, lui, marque normalement une option désactivée. Visiblement, il y a une inversion de l'action de ces deux codes. Mais le motif **(?-i)test** fonctionne comme prévu: il ne trouve que le mot tapé en minuscules.

Multiligne activé

Le code: **(?m)**. Le motif **(?m)^\w+** trouve un mot entier au début de chaque paragraphe de la chaîne de texte à examiner.

Qu'est-ce qu'une «regular expression»?

Une chaîne de caractères, formée de signes génériques (aux significations spécifiques) et de caractères littéraux. En bref, un motif utilisé pour rechercher dans des textes des chaînes de lettres distinctes, afin de les modifier.

Multiligne désactivé
Comme l'expression ci-devant, mais avec un trait d'union: **(?-m)**. Le motif **(?-m)^\w+** ne trouve que le premier mot de la chaîne de texte designée.

Ligne par ligne activé
InDesign insère **(?s)**. Cela signifie que tous les paragraphes seront traités comme une seule ligne. Ainsi, le motif **(?s)c.a** trouve les caractères «c» et «a» et le signe quelconque se trouvant entredeux.

Cela fonctionne également avec la chaîne «c<Retour>a».

Ligne par ligne désactivé
La recherche à l'aide du motif **(?-s)c.a** est limité à l'intérieur des paragraphes. Il y a une certaine parenté aux expressions multiligne, mais le résultat est différent.

Ignorer les espaces activé
Cette expression ne se trouve pas parmi les sous-articles des articles des «Caractères spéciaux pour la recherche». Mais on peut la mettre sans autre avec les «touches de modification».

Avec **(?x)** les espaces d'un motif sont ignorées. Cette option permet de structurer des motifs complexes, afin de les rendre plus lisibles. **(?x)a b c** trouve «abc» mais ne voit pas «a b c».

Ignorer les espaces désactivé
On peut insérer **(?-x)** dans la même ligne afin de rétablir le réglage par défaut après avoir activé l'option avec **(?x)**.

Ces deux expressions sont bel et bien expliquées sur le site web Adobe (voir le lien à la fin de cet article), mais différemment. De plus, à la place du «x» correct, les deux exemples contiennent un «s»...

Posix

Cette abréviation (fig. 3) veut dire *Portable Operating System Interface*, un lien entre une application et un système Unix.

Les articles offerts par InDesign CS3 pourraient être remplacés partiellement par des codes traditionnels. Mais cela rendrait les motifs moins savants...

[[:alnum:]]
Celui-ci sert à trouver des caractères alphabétiques ou numériques. Ainsi, des espaces, des signes de ponctuation, etc., sont exclus.

[[:alpha:]]
Cette expression limite la recherche aux signes alphabétiques. L'avantage: on ne doit pas définir des rangées comme **[a-z][A-Z]**.

[[:digit:]]
Pour trouver des chiffres quelconques. Il serait plus facile d'utiliser le **\d** présenté dans le premier article (ou une rangée comme **[0-9]**).

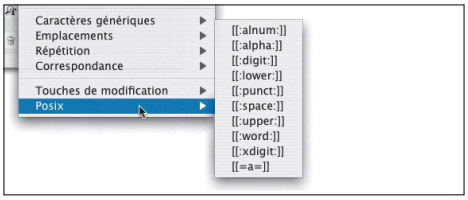

Fig. 3 – Certains des articles Posix pourraient être remplacés par des métacaractères traditionnels.

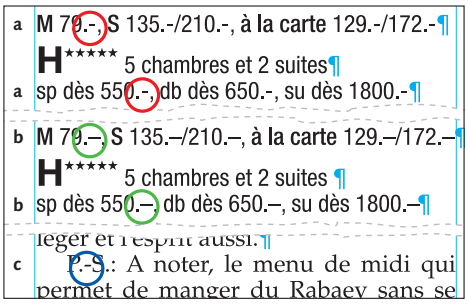

Fig. 4 – Les traits d'union des prix (a) devraient être modifiés en tirets demi-cadratin (b), mais sans changer les occurrences dans le texte courant (c).

[[:lower:]]
Il y avait une expression similaire dans le premier article. Celle-ci est l'abréviation de *lowercase*, le terme anglais pour minuscules.

[[:punct:]]
C'est déjà plus intéressant. Cette expression trouve des signes de ponctuation (angl. *punctuation*). Et non seulement des points et des virgules (ou des points d'interrogation ou d'exclamation) mais aussi des parenthèses, des barres de fraction et des traits d'union.

[[:space:]]
Ce terme anglais signifie «espace». Et ce sont exactement celles-ci qui sont trouvées, mais uniquement des espaces «normales». Les demi-cadratins et les huitièmes de cadratins sont exclues, ainsi que les espaces protégées.

[[:upper:]]
Le contraire de **[[:lower:]]**. Cette expression sert donc à trouver des caractères *uppercase*, des majuscules.

[[:word:]]
Tout le monde comprend ce terme anglais. Mais contrairement à ce que l'on pense, cette expression ne trouve pas de mots entiers. Seulement des caractères qui peuvent être contenus dans un mot, des signes alphanumériques. Ni des traits d'union ni des apostrophes.

[[:xdigit:]]
Cela trouve tous les caractères qui peuvent être contenus dans un nombre hexadécimal. Donc les chiffres 0 à 9 et les lettres abcdef ABCDEF.

Mais la recherche ne peut pas distinguer entre une valeur hexadécimale réelle (par exemple 0x3300FF) et un numéro dans une adresse postale (comme 77A). Il faudra inclure la distinction «0x» et le nombre dans le motif RegEx: **0x[[:xdigit:]]{6}**.

[[=a=]]
Cette expression trouve un caractère quelconque d'un certain jeu de glyphes. Le menu montre cet article comme l'intertitre ci-dessus. Mais ce n'est que **[[==]]** qui sera inséré dans le champ de la recherche. Il faut placer la lettre désirée entre les deux signes égal. **[[=a=]]** trouve, lors de la même action, «a», «à», «ä», «å», etc.

Pour terminer, un exemple de la pratique

Dans les textes fournis, le trait d'union est utilisé partout. Même aux endroits où il faudrait mettre un tiret demi-cadratin. Il est facile de les remplacer, dans le texte courant, par une routine traditionnelle: rechercher « - », remplacer par « – », rechercher « -, », remplacer par « –, ».

Mais il y a aussi les prix dans les informations (fig. 4). Rechercher «.-», remplacer par «.–» change les tirets des prix, mais également ceux du texte courant qu'il ne faudrait pas modifier (fig. 4c).

En utilisant **(?<=\d).-** il serait possible de faire la différence. Ce *Lookbehind* positif ne trouve que les signes «.-» précédés d'un chiffre quelconque. Ou bien rechercher **(\d)(.-)**, remplacer par **$1.–** serait également couronné de succès.

Mais la production tourne encore sous CS2. Donc, il n'est pas possible d'utiliser la recherche native d'InDesign. Le métacaractère **^9** trouve des chiffres quelconques, mais on ne peut pas insérer une référence dans le motif «remplacer». Contrairement au *Lookbehind*, le chiffre est sélectionné et il n'y a pas de possibilité de le remplacer par la même valeur.

On pourrait contourner ce problème: appliquer d'abord une couleur temporaire, ne changer que les traits en couleur, supprimer la couleur en la remplaçant par «Noir».

C'est beaucoup plus simple d'utiliser un script qui parcourt les paragraphes et ne fait le changement que si le nom de la feuille de style de paragraphe correspond.

Le dernier mot

Les «regular expressions» sont un complément d'InDesign CS3 très précieux. Elles vont certainement faciliter le travail. Mais les utilisateurs doivent investir pas mal de temps afin de se familiariser avec ces nouvelles possibilités.

Un point de départ valable: ouvrir l'aide d'InDesign. Entrer dans le champ rechercher le mot «GREP» et confirmer par retour. Les cinq résultats sont des liens menant à d'autres entrées de l'aide. Dans le premier chapitre il y a le lien suivant qui ouvre une page contenant des informations utiles.

www.adobe.com/go/learn_id_grep_fr