

Expressions régulières

Syntaxe des Expressions Régulières

La syntaxe des expressions régulières d'EmEditor est basée sur la syntaxe des expressions régulières de Perl.

Littéraux

Tous les caractères sont des littéraux sauf:

(".", "*", "?", "+", "(", ")", "{", "}", "[", "^[1]"], ">", "^[2]\", "^", "\$", "|", and "\".

Ces caractères sont des littéraux lorsqu'ils sont précédés par un "\". Un littéral est un caractère qui se trouve lui-même. Par exemple, la recherche de "\"?\" trouve chaque "\"" dans le document, ou la recherche de "\"Hello\" trouve chaque "\"Hello\" dans le document.

Metacaractères

Les tables suivantes contiennent la liste complète des metacaractères (non-littéraux) et leur comportement dans le cadre des expressions régulières.

\	Marque le caractère suivant comme caractère spécial, littéral, ou référence précédente. Par exemple, 'n' trouve le caractère "n". '\n' trouve le caractère SautdeLigne. La séquence '\\' trouve "\" et '\(' trouve "(".	e" trouve tout "e" qui commence une chaîne.		
\$	Trouve en position de fin de la chaîne d'entrée. Par exemple, "e\$" trouve tout "e" qui termine une chaîne.			
*	Trouve le caractère ou sous-expression qui le précède zéro ou plusieurs fois. Par exemple, zo* trouve "z" et "zoo". * est équivalent à \{0,\}.			
+	Trouve le caractère ou sous-expression qui le précède une ou plusieurs fois. Par exemple, \zo+' trouve "zo" et "zoo", mais pas "z". + est équivalent à \{1,\}.			
?	Trouve le caractère ou sous-expression qui le précède zéro ou une fois. Par exemple, "do(es)?" trouve le "do" dans "do" ou "does". ? est équivalent à \{0,1\}			
\{n\}	n est un entier non négatif. Trouve exactement n fois. Par exemple, \o\{2,\}' ne trouve pas le "o" de "Bob" mais trouve les deux "o" de "food".			
\{n,\}	n est un entier non négatif. Trouve au moins n fois. Par exemple, \o\{2,\}' ne trouve pas le "o" de "Bob" et trouve tous les "o" de "foooood". "o\{1,\}" est équivalent à \o+\'. \o\{0,\}' est équivalent à \o*\'.			

\{n,m\}	m et n sont des entiers non négatifs, où n = m. Trouve au moins n et au plus m fois. Par exemple, "o{1,3}" trouve les trois premiers "o" de "fooooood". Notez que vous ne pouvez pas mettre un espace entre la virgule et les nombres.			
?	Quand ce caractère suit immédiatement n\importe lequel des autres quantificateurs (*, +, ?, \{n\}, \{n,\}, \{n,m\}), la formule de recherche est non-avide. Une formule non-avide trouve le minimum possible des chaînes recherchées, alors que la formule avide par défaut trouve le maximum de chaînes recherchées. Par exemple, dans la chaîne "oooo", \'o+?\' trouve un simple "o", tandis que \'o+\' trouve tous les "o".			
.	Trouve tout caractère simple. Par exemple, ".e" trouvera un texte où n\importe quel caractère précède un "e", comme "he", "we", ou "me". Dans EmEditor Professionel, trouve un caractère SautdeLigne dans l'étendue spécifiée dans la boîte texte Lignes supp. recherche pour Expressions Régulières- mk:@MSITStore:C:\Program Files\EmEditor\mui36\emeditor.chm ^[3] /dlg/customize/search/lines_search.htm ^[4] si la case Les Expressions Régulières peuvent trouver les caractères SautdeLigne- mk:@MSITStore:C:\Program Files\EmEditor\mui36\emeditor.chm^[5] /dlg/customize/search/match_dot_nl.htm^[6] est cochée.			
(formule)	Les parenthèses ont deux usages: pour grouper une formule dans une sous-expression, et pour récupérer ce qu'a produit la recherche. Par exemple l'expression "(ab)*" trouvera toute la chaîne "ababab". Chaque sous-expression trouvée est capturée comme référence arrière (voir ci-dessous) numérotée de gauche à droite. Pour trouver les caractères parenthèses (), utilisez \'(\ ou \)\'. Les parenthèses ont deux usages: pour grouper une formule dans une sous-expression, et pour récupérer ce qu'a produit la recherche. Par exemple l'expression "(ab)*" trouvera toute la chaîne "ababab". Chaque sous-expression trouvée est capturée comme référence arrière (voir ci-dessous) numérotée de gauche à droite. Pour trouver les caractères parenthèses (), utilisez \'(\ ou \)\'.			
\1 - \9	Indique une référence arrière - une référence arrière est une référence à une sous-expression déjà trouvée précédemment. La référence concerne la sous-expression trouvée, non l'expression elle-même. Une référence arrière consiste en un caractère d'échappement "\" suivi d'un chiffre "1" à "9", "\1" pour se référer à la première sous-expression, "\2" à la seconde etc. Par exemple, "(a)\1" prendra "a" comme première référence arrière et trouvera tout texte "aa". Les références arrière peuvent aussi être utilisées dans la fonctionnalité Remplacer du menu Recherche . Utilisez les expressions régulières pour localiser un motif de texte, et le texte trouvé peut être remplacé par une référence arrière spécifiée. Par exemple, "(h)(e)" trouvera "he", et en mettant "\1" dans la boîte Remplacer Par remplacera "he" par "h" alors que "\2\1" remplacera "he" par "eh".			

(?:formule)	Une sous-expression qui répond à la formule mais ne capture pas ce qui est trouvé, c'est-à-dire que ce qui a été trouvé n'est pas stocké pour une éventuelle utilisation ultérieure avec des références arrière. Ceci est pratique pour combiner des parties d'une formule avec le caractère "ou" (\). Par exemple, \industr(?:y	ies) est une expression plus rapide que \industry\	industries\.
(?=formule)	Une sous-expression qui effectue une recherche vers l'avant de la présence de la formule à partir du point où une chaîne correspondante aura été trouvée. Par exemple, "x(=abc)" trouve un "x" seulement s'il est suivi de l'expression "abc". Ceci est une recherche sans capture, c'est-à-dire que ce qui a été trouvé n'est pas stocké pour une éventuelle utilisation ultérieure avec des références arrière. La formule ne peut pas contenir de caractère SautdeLigne.			
(?!pattern)	Une sous-expression qui effectue une recherche vers l'avant de la non présence de la formule à partir du point où une chaîne correspondante aura été trouvée. Par exemple, "x(!abc)" trouve un "x" seulement s'il n'est est suivi de l'expression "abc". Ceci est une recherche sans capture, c'est-à-dire que ce qui a été trouvé n'est pas stocké pour une éventuelle utilisation ultérieure avec des références arrière. La formule ne peut pas contenir de caractère SautdeLigne.			
(?=pattern)	Une sous-expression qui effectue une recherche vers l'arrière de la présence formule à partir du point où une chaîne correspondante aura été trouvée. Par exemple, "(=abc)x" trouve un "x" seulement s'il est précédé de l'expression "abc". Ceci est une recherche sans capture, c'est-à-dire que ce qui a été trouvé n'est pas stocké pour une éventuelle utilisation ultérieure avec des références arrière. La formule ne peut pas contenir de caractère SautdeLigne.			
(?	Une sous-expression qui effectue une recherche vers l'arrière de la non présence de la formule à partir du point où une chaîne correspondante aura été trouvée. Par exemple, "(?pression "abc". Ceci est une recherche sans capture, c'est-à-dire que ce qui a été trouvé n'est pas stocké pour une éventuelle utilisation ultérieure avec des références arrière. La formule ne peut pas contenir de caractère SautdeLigne.			
x\	y	Trouve ou x ou y. Par exemple, \z\	food\ trouve "z" ou "food". \'(z\	f)ood\ trouve "zood" ou "food".
\xyz ^[7]	Un ensemble de caractères. Trouve tout caractère contenu entre les crochets. Par exemple, \[abc ^[8] \ trouve le \a\ de "plain".			

\	xyz ^[19]	Un ensemble de caractères négatif. Trouve tout caractère non contenu entre les crochets. Par exemple, \ ^[10]	abc\ trouve le 'p' de "plain". ^[11]
\ ^[12] a-z ^[13]	Une fourchette de caractères. Trouve tout caractère qui est dans la fourchette. Par exemple, \ ^[14] a-z\ ^[14] trouve tout caractère alphanumérique minuscule qui est dans la fourchette 'a' à 'z'.		
\	a-z ^[15]	Une fourchette de caractères négative. Trouve tout caractère qui n'est pas dans la fourchette. Par exemple, \ ^[16]	a-z\ trouve tout caractère alphanumérique minuscule qui n'est pas dans la fourchette 'a' à 'z'. ^[17]

Classes de Caractères^[18]

Les classes de caractères suivantes sont utilisées dans un ensemble de caractères comme

"\^[19]:classname:\^[20]". Par exemple, "\:space:\^[21]" est l'ensemble de tous les caractères whitespace.

alnum	Tout caractère alphanumérique.
alpha	Tout caractère alphabétique a-z, A-Z, et tout autre caractère.
blank	Tout caractère blanc, soit un espace soit une tabulation.
cntrl	Tout caractère de contrôle.
digit	Tout chiffre 0-9.
graph	Tout caractère graphique.
lower	Tout caractère minuscule a-z, et tout autre caractère minuscule.
print	Tout caractère imprimable.
punct	Tout signe de ponctuation.
space	Tout caractère whitespace.
upper	Tout caractère majuscule a-z, et tout autre caractère majuscule.
xdigit	Tout chiffre hexadécimal, 0-9, a-f et A-F.
word	Tout caractère de mot - tous les caractères alphanumériques plus le tiret de soulignement.

unicode	Tout caractère dont le code est plus grand que 255.
---------	---

Séquences d'échappement à caractère simple

Les séquences d'échappement suivantes sont des alias pour les caractères simples:

0x07	\a	Caractère Bell.
0x0C	\f	Saut de page.
0x0A	\n	Saut de ligne.
0x0D	\r	Retour chariot.
0x09	\t	Caractère de tabulation.
0x0B	\v	Tabulation verticale.
0x1B	\e	Caractère d'échappement ASCII.
Odd	\Odd	Un caractère codé en octal, où dd est un chiffre octal ou plus.
0xXX	\xXX	Un caractère codé en hexadécimal, où XX est un chiffre hexadécimal ou plus (un caractère Unicode).
0xXXXX	\x{XXXX}	Un caractère codé en hexadécimal, où XXXX est un chiffre hexadécimal ou plus (un caractère Unicode).
Z-\'@\'	\cZ Z-\'@\'	Une séquence d'échappement ASCII control-Z, où Z est un caractère ASCII plus grand ou égal au code du caractère '\'@\'

Classes de caractères séquences d'échappement

Les séquences d'échappement suivantes peuvent être utilisées pour représenter entièrement les classes de caractères:

\w	Tout caractère de mot - tous les caractères alphanumériques plus le tiret de soulignement.
\W	Complément de \w - trouve tout non-caractère de mot
\s	Tout caractère whitespace.
\S	Complément de \s.
\d	Tout chiffre 0-9.
\D	Complément de \d.
\l	Tout caractère minuscule a-z.
\L	Complément de \l.
\u	Tout caractère majuscule A-Z.
\U	Complément de \u.
\C	Tout caractère simple, équivalent à '\.\'

\Q	L'opérateur début quote, tout ce qui suit est traité comme caractère littéral jusqu'à ce qu'un caractère fin quote \E soit trouvé.
\E	L'opérateur fin quote, termine une séquence commencée avec \Q.

Expressions de Remplacement

Les expressions suivantes sont disponibles pour la boîte **Remplacer Par** de la boîte de dialogue **Remplacer** et dans la boîte de dialogue **Remplacer dans Fichiers**.

\0	Indique une référence arrière pour la totalité de l'expression régulière.
\1 - \9	Indique une référence arrière - une référence arrière est une référence à une sous-expression qui a déjà été trouvée précédemment. La référence concerne la sous-expression trouvée, non l'expression elle-même. Une référence arrière consiste en un caractère d'échappement "\" suivi d'un chiffre "1" à "9", "\1" pour se référer à la première sous-expression, "\2" à le seconde etc.
\n	Un saut de ligne.
\r	Un retour chariot dans le cas de Remplacer dans Fichiers . Voir aussi Pour spécifier Nouvelles Lignes-mk:@MSITStore:C:\Program Files\EmEditor\mui36\emeditor.chm::/howto/search/search_nl.htm ^[22] .
\t	Une tabulation
\L	Force tous les caractères postérieurs en minuscules.
\U	Force tous les caractères postérieurs en majuscules.
\H	Force tous les caractères postérieurs en demi-largeur.
\F	Force tous les caractères postérieurs en pleine-largeur.
\E	Annule les précédents \L, \U, \F, ou \H. (EmEditor Professional only)

Notes

1. Dans **Rechercher dans Fichiers** et dans **Remplacer dans Fichiers**, le retour chariot (\r) et le saut de ligne (\n) doivent être remplacés avec prudence. Voir Pour Sécifier Sauts de Ligne-mk:@MSITStore:C:\Program Files\EmEditor\mui36\emeditor.chm::/howto/search/search_nl.htm^[23] pour les détails.
2. Afin de pouvoir utiliser certaines séquences d'échappement dans EmEditor, comme "\l", "\u" et leurs compléments, l'option Respecter la Casse doit être cochée.

^[1] <https://outilingua.net/>

^[2] <https://outilingua.net/>

^[3] <https://outilingua.net/>

^[4] <https://outilingua.net/>

^[5] <https://outilingua.net/>

^[6] <https://outilingua.net/>

^[7] <https://outilingua.net/xyz%5C>

- [8] <https://outilingua.net/abc%5C>
- [9] <https://outilingua.net/%5C>
- [10] <https://outilingua.net/%5C>
- [11] <https://outilingua.net/%5C>
- [12] <https://outilingua.net/%5C>
- [13] <https://outilingua.net/a-z%5C>
- [14] <https://outilingua.net/a-z%5C>
- [15] <https://outilingua.net/%5C>
- [16] <https://outilingua.net/%5C>
- [17] <https://outilingua.net/%5C>
- [18] <https://outilingua.net/%5C>
- [19] <https://outilingua.net/%5C>
- [20] <https://outilingua.net/%3Aclassname%3A%5C>
- [21] <https://outilingua.net/%3Aspace%3A%5C>
- [22] <https://outilingua.net/>
- [23] <https://outilingua.net/>