

Cet article a fait l'objet d'une traduction automatique. Déplacez votre pointeur sur les phrases de l'article pour voir la version originale de ce texte. [Informations supplémentaires.](#)

☒ Traduction ☐ Source

# Langage des expressions régulières - Aide-mémoire

**.NET Framework 4.5** 1 sur 1 ont trouvé cela utile

Une expression régulière est un modèle que le moteur des expressions régulières tente de faire correspondre dans le texte d'entrée. Un modèle se compose d'un ou de plusieurs littéraux de caractère, opérateurs ou constructions. Pour obtenir une brève présentation, consultez [Expressions régulières du .NET Framework](#).

Chaque section dans cette référence rapide répertorie une catégorie particulière de caractères, d'opérateurs et de constructions que vous pouvez utiliser pour définir des expressions régulières :

- [Caractères d'échappement](#)
- [Classes de caractères](#)
- [Ancres](#)
- [Constructions de regroupement](#)
- [Quantificateurs](#)
- [Constructions de backreference](#)
- [Constructions d'alternative](#)
- [Substitutions](#)
- [Options d'expressions régulières](#)
- [Constructions diverses](#)

## Caractères d'échappement

La barre oblique inverse (\) dans une expression régulière indique que le caractère qui le suit est un caractère spécial (comme indiqué dans le tableau suivant), ou qu'il doit être interprété littéralement. Pour plus d'informations, consultez [Caractères d'échappement dans les expressions régulières](#).

Caractère d'échappement	Description	Modèle	Correspondance
<code>\a</code>	Correspond à un caractère de cloche, \u0007.	<code>\a</code>	"\u0007" dans "Erreur !" + '\u0007'
<code>\b</code>	Dans une classe de caractères, correspond à un retour arrière, \u0008.	<code>[\b]{3,}</code>	"\b\b\b\b" dans "\b\b\b\b"
<code>\t</code>	Correspond à une tabulation, \u0009.	<code>(\w+)\t</code>	"item1\t", "item2\t" dans "item1\titem2\t"

<code>\r</code>	Correspond à un retour chariot, \u000D. ( <code>\r</code> n'est pas équivalent au caractère de saut de ligne, <code>\n</code> .)	<code>\r\n(\w+)</code>	" <code>\r\nCe</code> " dans " <code>\r\nCe sont\ndeux lignes.</code> "
<code>\v</code>	Correspond à une tabulation verticale, \u000B.	<code>[\v]{2,}</code>	" <code>\v\v\v</code> " dans " <code>\v\v\v</code> "
<code>\f</code>	Correspond à un saut de page, \u000C.	<code>[\f]{2,}</code>	" <code>\f\f</code> " dans " <code>\f\f</code> "
<code>\n</code>	Correspond à une nouvelle ligne, \u000A.	<code>\r\n(\w+)</code>	" <code>\r\nCe</code> " dans " <code>\r\nCe sont\ndeux lignes.</code> "
<code>\e</code>	Correspond à un caractère d'échappement, \u001B.	<code>\e</code>	" <code>\x001B</code> " dans " <code>\x001B</code> "
<code>\nnn</code>	Utilise la représentation octale pour spécifier un caractère ( <i>nnn</i> se compose de deux ou trois chiffres).	<code>\w\040\w</code>	"a b", "c d" dans "a bc d"
<code>\x nn</code>	Utilise une représentation hexadécimale pour spécifier un caractère ( <i>nn</i> se compose de deux chiffres exactement).	<code>\w\x20\w</code>	"a b", "c d" dans "a bc d"
<code>\c X</code> <code>\c x</code>	Correspond au caractère de commande ASCII spécifié par <i>X</i> ou <i>x</i> , où <i>X</i> ou <i>x</i> représente la lettre du caractère de contrôle.	<code>\cC</code>	" <code>\x0003</code> " dans " <code>\x0003</code> " (Ctrl-C)
<code>\u nnnn</code>	Correspond à un caractère Unicode en utilisant la représentation hexadécimale (quatre chiffres exactement, représentés par <i>nnnn</i> ).	<code>\w\u0020\w</code>	"a b", "c d" dans "a bc d"
<code>\</code>	Lorsque ce caractère d'échappement est suivi d'un caractère non identifié comme caractère d'échappement, correspond au caractère lui-même. Par exemple, <code>\*</code> est identique à <code>\x2A</code> , et <code>\.</code> est identique à <code>\x2E</code> . Cela permet au moteur des expressions régulières de lever l'ambiguïté d'éléments de langage (tels que <code>*</code> ou <code>?</code> ) et de caractères littéraux (représentés par <code>\*</code> or <code>\?</code> ).	<code>\d+[\+-x\*]\d+</code> <code>\d+[\+-x\*\d+</code>	"2+2" et "3*9" dans "(2+2) * 3*9"

[Retour au début](#)

## Classes de caractères

Une classe de caractères fait correspondre tout caractère d'un jeu de caractères. Les classes de caractères incluent les éléments de langage répertoriés dans le tableau suivant. Pour plus d'informations, consultez [Classes de caractères dans les expressions régulières](#).

Classe de caractères	Description	Modèle	Correspondance
<code>[ character_group ]</code>	Correspond à n'importe quel caractère unique de <i>character_group</i> . Par défaut, la correspondance respecte la casse.	<code>[ae]</code>	"a" dans "gras" "a", "e" dans "laine"
<code>[^ character_group ]</code>	Négation : correspond à tout caractère unique qui ne se trouve pas dans <i>character_group</i> . Par défaut, les	<code>[^aei]</code>	"r", "g", "n" dans "règne"

	caractères de <i>character_group</i> respectent la casse.		
[ <i>first</i> - <i>last</i> ]	Plage de caractères : correspond à n'importe quel caractère unique dans la plage comprise entre <i>first</i> et <i>last</i> .	[A-Z]	"A", "B" dans "AB123"
.	Caractère générique : correspond à tout caractère à l'exception de \n. Pour faire correspondre un caractère littéral « point » (. ou \u002E), vous devez le faire précéder du caractère d'échappement (\.).	a.e	"ave" dans "navet" "ate" dans "plate"
\p{ <i>name</i> }	Correspond à tout caractère unique de la catégorie générale Unicode ou du bloc nommé spécifié par <i>name</i> .	\p{Lu} \p{IsCyrillic}	"M", "G" dans "Mardi Gras" "Д", "Ж" dans "ДЖем"
\P{ <i>name</i> }	Correspond à tout caractère unique qui ne se trouve pas dans la catégorie générale Unicode ou le bloc nommé spécifié par <i>name</i> .	\P{Lu} \P{IsCyrillic}	"a", "r", "d", "i" dans "Mardi" "e", "m" dans "ДЖем"
\w	Correspond à n'importe quel caractère alphabétique.	\w	"I", "D", "A", "1", "3" dans "ID A1.3"
\W	Correspond à tout caractère autre qu'un caractère de mot.	\W	" ", "." dans "ID A1.3"
\s	Correspond à tout caractère espace blanc.	\w\s	"D " dans "ID A1.3"
\S	Correspond à tout caractère autre qu'un espace blanc.	\s\S	" _" dans "int __ctr"
\d	Correspond à n'importe quel chiffre décimal.	\d	"4" dans "4 = IV"
\D	Correspond à n'importe quel caractère autre qu'un chiffre décimal.	\D	" ", "=", " ", "I", "V" dans "4 = IV"

[Retour au début](#)

## Ancres

Les ancres, ou assertions de largeur nulle atomiques, entraînent le succès ou l'échec d'une correspondance en fonction de la position actuelle dans la chaîne, mais elles n'entraînent pas l'avancement du moteur à travers la chaîne ou la consommation de caractères. Les métacaractères répertoriés dans le tableau suivant sont des ancres. Pour plus d'informations, consultez [Ancres dans les expressions régulières](#).

Assertion	Description	Modèle	Correspondance
^	La correspondance doit commencer au début de la chaîne ou de la ligne.	^\d{3}	"901" dans "901-333-"
\$	La correspondance doit se produire à la fin de la chaîne ou avant \n à la fin de la ligne ou de la chaîne.	-\d{3}\$	"-333" dans "-901-333"

<b>\A</b>	La correspondance doit se produire au début de la chaîne.	<b>\A\d{3}</b>	"901" dans "901-333-"
<b>\Z</b>	La correspondance doit se produire à la fin de la chaîne ou avant <b>\n</b> à la fin de la chaîne.	<b>-\d{3}\Z</b>	"-333" dans "-901-333"
<b>\z</b>	La correspondance doit se produire à la fin de la chaîne.	<b>-\d{3}\z</b>	"-333" dans "-901-333"
<b>\G</b>	La correspondance doit se produire à l'emplacement où la correspondance précédente s'est terminée.	<b>\G\(\d\)</b>	"(1)", "(3)", "(5)" dans "(1)(3)(5)[7](9)"
<b>\b</b>	La correspondance doit se produire sur une limite entre un caractère <b>\w</b> (alphanumérique) et un caractère <b>\W</b> (non alphanumériques).	<b>\bw+\s\w+\b</b>	"them theme", "them them" dans "them theme them them"
<b>\B</b>	La correspondance ne doit pas se produire sur une limite <b>\b</b> .	<b>\Bend\w*\b</b>	"ends", "ender" dans "end sends endure lender"

[Retour au début](#)

## Constructions de regroupement

Les constructions de regroupement délimitent les sous-expressions d'une expression régulière et capturent généralement les sous-chaînes d'une chaîne d'entrée. Les constructions de regroupement incluent les éléments de langage répertoriés dans le tableau suivant. Pour plus d'informations, consultez [Constructions de regroupement dans les expressions régulières](#).

Construction de regroupement	Description	Modèle	Correspondance
<b>( <i>subexpression</i> )</b>	Capture la sous-expression mise en correspondance et lui assigne un nombre ordinal de base zéro.	<b>(\w)\1</b>	"oo" dans "boot"
<b>(?&lt; <i>name</i> &gt; <i>subexpression</i> )</b>	Capture la sous-expression mise en correspondance dans un groupe nommé.	<b>(?&lt;double&gt;\w)\k&lt;double&gt;</b>	"oo" dans "boot"
<b>(?&lt; <i>name1</i> - <i>name2</i> &gt; <i>subexpression</i> )</b>	Définit un équilibre de définition de groupe. Pour plus d'informations, consultez la section « Équilibre de définition de groupe » dans <a href="#">Constructions de regroupement dans les expressions régulières</a> .	<b>((? 'Open' \()[^\(\)]*)+((? 'Close-Open' \()[^\(\)]*)+)*(?(Open)?!))\$</b>	"((1-3)*(3-1))" dans "3+2^((1-3)*(3-1))"
<b>(?: <i>subexpression</i> )</b>	Définit un groupe sans capture.	<b>Write(?:Line)?</b>	"WriteLine" dans "Console.WriteLine()"
<b>(?imnsx-imnsx: <i>subexpression</i> )</b>	Active ou désactive les options spécifiées au sein de <i>subexpression</i> . Pour plus d'informations, consultez <a href="#">Options des expressions régulières</a> .	<b>A\d{2}(?i:\w+)\b</b>	"A12xl", "A12XL" dans "A12xl A12XL a12xl"
<b>(?= <i>subexpression</i> )</b>	Assertion de préanalyse positive de largeur nulle.	<b>\w+(?=\. )</b>	"est", "courait", et "couché" dans "Il est. Le chien courait. Le soleil est couché."

<b>(?! <i>subexpression</i> )</b>	Assertion de préanalyse négative de largeur nulle.	<b>\b(?!un)\w+\b</b>	"retourner", "utilisée" dans "retourner une unité utilisée"
<b>(?&lt;= <i>subexpression</i> )</b>	Assertion de postanalyse positive de largeur nulle.	<b>(?&lt;=19)\d{2}\b</b>	"99", "50", "05" dans "1851 1999 1950 1905 2003"
<b>(?&lt;! <i>subexpression</i> )</b>	Assertion de postanalyse négative de largeur nulle.	<b>(?&lt;!19)\d{2}\b</b>	"51", "03" dans "1851 1999 1950 1905 2003"
<b>(?&gt; <i>subexpression</i> )</b>	Sous-expressions de recherche non rétroactive (ou « gourmandes »).	<b>[13579](?&gt;A+B+)</b>	"1ABB", "3ABB" et "5AB" dans "1ABB 3ABBC 5AB 5AC"

[Retour au début](#)

## Quantificateurs

Un quantificateur indique combien d'instances de l'élément précédent (qui peut être un caractère, un groupe ou une classe de caractères) doivent être présentes dans la chaîne d'entrée pour qu'il y ait correspondance. Les quantificateurs incluent les éléments de langage répertoriés dans le tableau suivant. Pour plus d'informations, consultez [Quantificateurs dans les expressions régulières](#).

Quantifieur	Description	Modèle	Correspondance
<b>*</b>	Correspond zéro, une ou plusieurs fois à l'élément précédent.	<b>\d*\.\d</b>	".0", "19.9", "219.9"
<b>+</b>	Correspond une ou plusieurs fois à l'élément précédent.	<b>"be+"</b>	"bee" dans "beefsteak", "be" dans "besoin"
<b>?</b>	Correspond zéro ou une fois à l'élément précédent.	<b>"rai?n"</b>	"rang", "train"
<b>{ <i>n</i> }</b>	Correspond à l'élément précédent exactement <i>n</i> fois.	<b>",\d{3}"</b>	",043" dans "1,043.6", ",876", ",543" et ",210" dans "9,876,543,210"
<b>{ <i>n</i> , }</b>	Correspond à l'élément précédent au moins <i>n</i> fois.	<b>"\d{2,}"</b>	"166", "29", "1930"
<b>{ <i>n</i> , <i>m</i> }</b>	Correspond à l'élément précédent au moins <i>n</i> fois, mais pas plus de <i>m</i> fois.	<b>"\d{3,5}"</b>	"166", "17668" "19302" dans "193024"
<b>*?</b>	Correspond zéro fois ou plus à l'élément précédent, mais le moins de fois possible.	<b>\d*?\.\d</b>	".0", "19.9", "219.9"
<b>+?</b>	Correspond une ou plusieurs fois à l'élément précédent, mais le moins de fois possible.	<b>"be+?"</b>	"bee" dans "beefsteak", "be" dans "besoin"
<b>??</b>	Correspond zéro ou une fois à l'élément précédent, mais le moins de fois possible.	<b>"rai??n"</b>	"rang", "train"
<b>{ <i>n</i> }?</b>	Correspond exactement <i>n</i> fois à l'élément précédent.	<b>",\d{3}?"</b>	",043" dans "1,043.6", ",876", ",543" et ",210" dans "9,876,543,210"

<b><math>\{ n , \} ?</math></b>	Correspond au moins $n$ fois à l'élément précédent, mais le moins de fois possible.	<code>"\d{2,}?"</code>	"166", "29", "1930"
<b><math>\{ n , m \} ?</math></b>	Correspond entre $n$ et $m$ fois à l'élément précédent, mais le moins de fois possible.	<code>"\d{3,5}?"</code>	"166", "17668" "193", "024" dans "193024"

[Retour au début](#)

## Constructions de backreference

Une backreference permet qu'une sous-expression précédemment mise en correspondance soit ensuite identifiée dans la même expression régulière. Le tableau suivant répertorie les constructions de backreference prises en charge par les expressions régulières dans le .NET Framework. Pour plus d'informations, consultez [Constructions de backreference dans les expressions régulières](#).

Construction de backreference	Description	Modèle	Correspondance
<code>\ number</code>	Backreference. Correspond à la valeur d'une sous-expression numérotée.	<code>(\w)\1</code>	"ee" dans "seek"
<code>\k&lt; name &gt;</code>	Backreference nommée. Correspond à la valeur d'une expression nommée.	<code>(?&lt;char&gt;\w) \k&lt;char&gt;</code>	"ee" dans "seek"

[Retour au début](#)

## Constructions d'alternative

Les constructions d'alternative modifient une expression régulière pour permettre la correspondance de type inclusif/exclusif. Ces constructions incluent les éléments de langage répertoriés dans le tableau suivant. Pour plus d'informations, consultez [Constructions d'alternative dans les expressions régulières](#).

Construction d'alternative	Description	Modèle	Correspondance
<code> </code>	Correspond à tout élément séparé par le caractère barre verticale ( ).	<code>th(e is at)</code>	"the", "this" dans "this is the day. "
<code>(?( expression ) yes   no )</code>	Correspond à <i>yes</i> si le modèle d'expression régulière indiqué par <i>expression</i> correspond ; sinon, correspond à <i>no</i> (facultatif). <i>expression</i> est interprétée comme une assertion de largeur nulle.	<code>(?(A)\d{2} \b \b\d{3}\b)</code>	"A10", "910" dans "A10 C103 910"
<code>(?( name ) yes   no )</code>	Correspond à <i>yes</i> si <i>name</i> , un groupe de capture nommé ou numéroté, a une correspondance ; sinon, correspond à <i>no</i> (facultatif).	<code>(?&lt;quoted&gt;)? (?(quoted).+?) \S+\s)</code>	Dogs.jpg, "Yiska playing.jpg" dans "Dogs.jpg "Yiska playing.jpg""

[Retour au début](#)

## Substitutions

Les substitutions sont des éléments de langage d'expression régulière pris en charge dans les modèles de remplacement. Pour plus d'informations, consultez [Substitutions dans les expressions régulières](#). Les métacaractères répertoriés dans le tableau suivant sont des assertions de largeur nulle atomiques.

Caractère	Description	Modèle	Modèle de remplacement	Chaîne d'entrée	Chaîne de résultat
<code>\$number</code>	Remplace la sous-chaîne mise en correspondance par le groupe <i>number</i> .	<code>\b(\w+)(\s)(\w+)\b</code>	<code>\$3\$2\$1</code>	"un deux"	"deux un"
<code>\${name}</code>	Remplace la sous-chaîne mise en correspondance par le groupe nommé <i>name</i> .	<code>\b(&lt;word1&gt;\w+)(\s)(?&lt;word2&gt;\w+)\b</code>	<code>\${word2}\${word1}</code>	"un deux"	"deux un"
<code>\$\$</code>	Remplace un "\$" littéral.	<code>\b(\d+)\s?USD</code>	<code>\$\$\$1</code>	"103 USD"	"\$103"
<code>\$&amp;</code>	Remplace une copie de la totalité de la correspondance.	<code>(\\$(\d*(\.\d+)?)\d*){1}</code>	<code>**\$&amp;</code>	"\$1.30"	"**\$1.30**"
<code>\$`</code>	Remplace tout le texte de la chaîne d'entrée avant la correspondance.	<code>B+</code>	<code>\$`</code>	"AABBCC"	"AAAACC"
<code>\$'</code>	Remplace tout le texte de la chaîne d'entrée après la correspondance.	<code>B+</code>	<code>\$'</code>	"AABBCC"	"AACCCC"
<code>\$+</code>	Remplace le dernier groupe qui a été capturé.	<code>B+(C+)</code>	<code>\$+</code>	"AABBCCDD"	AACCDD
<code>\$_</code>	Remplace la chaîne d'entrée entière.	<code>B+</code>	<code>\$_</code>	"AABBCC"	"AAAABBCCCC"

[Retour au début](#)

## Options des expressions régulières

Vous pouvez définir des options qui contrôlent comment le moteur des expressions régulières interprète un modèle d'expression régulière. Bon nombre de ces options peuvent être spécifiées inline (dans le modèle d'expression régulière) ou sous forme d'une ou de plusieurs constantes [RegexOptions](#). Cette référence rapide répertorie uniquement les options inline. Pour plus d'informations sur les options inline et [RegexOptions](#), consultez l'article [Options des expressions régulières](#).

Vous pouvez spécifier une option inline de deux façons :

- Avec [miscellaneous construct](#)(**?imsnx-imnsx**), où un signe moins (-) devant une option ou un jeu d'options désactive ces options. Par exemple, (**?i-mn**) active la correspondance qui ne respecte pas la casse (**i**), désactive le mode multiligne(**m**) et désactive les captures de groupe sans nom (**n**). L'option s'applique au modèle d'expression régulière depuis le point au niveau duquel l'option est définie et est effective jusqu'à la fin du modèle ou jusqu'au point au niveau duquel une autre construction inverse l'option.
- Avec [grouping construct](#)(**?imsnx-imnsx:subexpression**), qui définit des options uniquement pour le groupe spécifié.

Le moteur des expressions régulières du .NET Framework prend en charge les options inline suivantes.

Option	Description	Modèle	Correspondance
<b>i</b>	Utilisez la correspondance qui ne respecte pas la casse	<code>\b(?:i)a(?:-i)a\w+\b</code>	« aardvark », « aaaAuto » dans « aardvark AAAuto aaaAuto Adam breakfast »
<b>m</b>	Utilisez le mode multiligne. <code>^</code> et <code>\$</code> correspondent au début et à la fin d'une ligne, au lieu du début et de la fin d'une chaîne.	Pour obtenir un exemple, consultez la section « Mode multiligne » dans <a href="#">Options des expressions régulières</a> .	
<b>n</b>	Ne capturez pas de groupes sans nom.	Pour obtenir un exemple, consultez la section « Captures explicites uniquement » dans <a href="#">Options des expressions régulières</a> .	
<b>s</b>	Utilisez le mode à ligne simple.	Pour obtenir un exemple, consultez la section « Mode sur une ligne » dans <a href="#">Options des expressions régulières</a> .	
<b>x</b>	Ignorez l'espace blanc sans séquence d'échappement dans le modèle d'expression régulière.	<code>\b(?:x) \d+ \s \w+</code>	« 1 aardvark », « 2 cats » dans « 1 aardvark 2 cats IV centurions »

[Retour au début](#)

## Constructions diverses

Diverses constructions modifient un modèle d'expression régulière ou fournissent des informations le concernant. Le tableau suivant répertorie les diverses constructions prises en charge par le .NET Framework. Pour plus d'informations, consultez [Constructions diverses dans les expressions régulières](#).

Construction	Définition	Exemple
<b>(?imnsx-imnsx)</b>	Active ou désactive des options telles que le non-respect de la casse au milieu d'un modèle. Pour plus d'informations, consultez <a href="#">Options des expressions régulières</a> .	<code>\bA(?:i)b\w+\b</code> correspond à "ABA", "Able" dans "ABA Able Act"
<b>(?# comment )</b>	Commentaire inline. Le commentaire se termine à la première parenthèse fermante.	<code>\bA(?:#Matches words starting with A)\w+\b</code>
<b># [to end of line]</b>	Commentaire en mode X. Le commentaire commence au caractère <code>#</code> sans séquence d'échappement et se poursuit jusqu'à la fin de la ligne.	<code>(?:x)\bA\w+\b#Matches words starting with A</code>

[Retour au début](#)

## Voir aussi

### Référence

[System.Text.RegularExpressions](#)[Regex](#)

### Concepts

[Classes d'expressions régulières](#)[Autres ressources](#)



[Expressions régulières du .NET Framework](#)  
[Exemples d'expressions régulières](#)

---

## Ajouts de la communauté

---

© 2014 Microsoft. Tous droits réservés.