

RÉFÉRENCE RAPIDE EXPRESSIONS RÉGULIÈRES

Motifs, quantificateurs, groupes, assertions, drapeaux

Motifs de base

Métacaractères
<code>.</code> N'importe quel caractère (sauf saut de ligne)
<code>^</code> Début de chaîne / ligne
<code>\$</code> Fin de chaîne / ligne
<code>*</code> 0 fois ou plus ce qui précède
<code>+</code> 1 fois ou plus ce qui précède
<code>?</code> 0 ou 1 fois ce qui précède (optionnel)
<code>\</code> Échapper un métacaractère

Correspondance littérale

<code>hello</code>	<code># matches "hello" exactly</code>
<code>a.c</code>	<code># matches "abc", "alc", "a-c", etc.</code>
<code>\.txt</code>	<code># matches literal ".txt"</code>

Classes de caractères

Expressions entre crochets

<code>[abc]</code>	Correspond à a, b ou c
<code>[^abc]</code>	Tout sauf a, b, c
<code>[a-z]</code>	Lettre minuscule
<code>[A-Z]</code>	Lettre majuscule
<code>[0-9]</code>	Chiffre
<code>[a-zA-Z0-9]</code>	Alphanumérique

Classes abrégées

<code>\d</code>	Chiffre <code>[0-9]</code>
<code>\D</code>	Non-chiffre <code>[^0-9]</code>
<code>\w</code>	Caractère de mot <code>[a-zA-Z0-9_]</code>
<code>\W</code>	Caractère non-mot
<code>\s</code>	Espace blanc <code>[\t\n\r\f]</code>
<code>\S</code>	Non-espace blanc

Quantificateurs

Quantificateurs gourmands

<code>*</code>	0 fois ou plus (gourmand)
<code>+</code>	1 fois ou plus (gourmand)
<code>?</code>	0 ou 1 fois (gourmand)
<code>{n}</code>	Exactement n fois
<code>{n,}</code>	n fois ou plus
<code>{n,m}</code>	Entre n et m fois

Quantificateurs paresseux

<code>*?</code>	0 fois ou plus (paresseux / non gourmand)
<code>+?</code>	1 fois ou plus (paresseux)
<code>??</code>	0 ou 1 fois (paresseux)
<code>{n,m}?</code>	Entre n et m fois (paresseux)

Les quantificateurs paresseux correspondent au minimum de caractères possible

Gourmand vs Paresseux

<code><.+></code>	<code># greedy: "bold"</code>
<code><.+?></code>	<code># lazy: ""</code>

Ancre

<code>^</code>	Début de chaîne (ou de ligne avec le drapeau <code>m</code>)
<code>\$</code>	Fin de chaîne (ou de ligne avec le drapeau <code>m</code>)
<code>\b</code>	Limite de mot
<code>\B</code>	Non-limite de mot
<code>\A</code>	Début de chaîne (non affecté par <code>m</code>)
<code>\Z</code>	Fin de chaîne (non affecté par <code>m</code>)

Exemples d'ancres

<code>"hello"</code>	<code># starts with "Hello"</code>
<code>world\$</code>	<code># ends with "world"</code>
<code>\bword\b</code>	<code># "word" as whole word</code>
<code>\Bword\b</code>	<code># "word" inside another word</code>

Groupes & Alternance

Groupes capturants

<code>(abc)</code>	<code># capture group: match "abc"</code>
<code>(a b c)</code>	<code># alternation: a or b or c</code>
<code>(cat dog)</code>	<code># match "cat" or "dog"</code>
<code>(\d{3})-(\d{4})</code>	<code># groups: "123-4567"</code>

Types de groupe

<code>(pattern)</code>	Groupe capturant
<code>(?pattern)</code>	Groupe non capturant
<code>(?P<name>pat)</code>	Groupe nommé (Python)
<code>(?<name>pat)</code>	Groupe nommé (JS, .NET)
<code>\1 \2</code>	Références aux groupes 1, 2
<code>a b</code>	Alternance : a ou b

Assertion avant & arrière

<code>(?=pattern)</code>	Assertion avant positive
<code>(?!pattern)</code>	Assertion avant négative
<code>(?<=pattern)</code>	Assertion arrière positive
<code>(?<!pattern)</code>	Assertion arrière négative

Exemples d'assertions

<code>\d+(?= USD)</code>	<code># digits followed by " USD"</code>
<code>\d+(?! USD)</code>	<code># digits NOT followed by " USD"</code>
<code>(?<=\\$)\d+</code>	<code># digits preceded by "\$"</code>
<code>(?!\\$)\d+</code>	<code># digits NOT preceded by "\$"</code>

Les assertions correspondent à une position sans consommer de caractères

Motifs courants

<code>\d{1,3}\.\d{1,3}\d{3}</code>	Adresse IPv4 (basique)
<code>[\w.-]+@[\w.-]+\.[\w.-]+</code>	Email (basique)
<code>https://[\w.-]+/?&#=[\w.-]+</code>	URL (basique)
<code>\(\d{3}\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}</code>	Numéro de téléphone US
<code>\d{4}-\d{2}-\d{2}</code>	Date (AAAA-MM-JJ)
<code>#?[0-9a-f]{6}</code>	Code couleur hexadécimal

Ces motifs sont simplifiés ; la production peut nécessiter une validation plus stricte

Drapeaux

<code>g</code>	Global ; trouver toutes les correspondances
<code>i</code>	Insensible à la casse
<code>m</code>	Multiligne ; <code>^</code> / <code>\$</code> correspondent aux limites de ligne
<code>s</code>	Dotall : <code>.</code> correspond aussi aux sauts de ligne
<code>x</code>	Verbeux : ignorer les espaces, autoriser les commentaires
<code>u</code>	Unicode : support Unicode complet

Utilisation des drapeaux par langage

<code>/pattern/gi</code>	<code># JavaScript</code>
<code>re.compile("pat", re.I re.M)</code>	<code># Python</code>
<code>grep -IE "pattern"</code>	<code># grep (extended)</code>