

# Regex

Quelques explications et exemples sur les expressions régulières et leurs utilisations.

- [\[Regex\] Kit de base](#)
- [\[Regex\] Exemples](#)

# [Regex] Kit de base

## Introduction

Les **expressions régulières**, souvent appelées "**regex**" ou "**regexp**", sont des motifs de recherche ou de correspondance utilisés pour trouver des motifs dans des chaînes de caractères.

•[RegEx]\*

## Métacaractères de base :

- `.` : Correspond à n'importe quel caractère, sauf une nouvelle ligne.
- `*` : Correspond à zéro, une ou plusieurs occurrences du caractère précédent.
- `+` : Correspond à une ou plusieurs occurrences du caractère précédent.
- `?` : Correspond à zéro ou une occurrence du caractère précédent.
- `|` : Permet de spécifier des alternatives. Par exemple, `chat|chien` correspondra à "chat" ou "chien" dans un texte.

## Classes de caractères :

- `[abc]` : Correspond à l'un des caractères a, b ou c.
- `[a-z]` : Correspond à n'importe quelle lettre minuscule.
- `[A-Z]` : Correspond à n'importe quelle lettre majuscule.
- `[0-9]` : Correspond à n'importe quel chiffre.
- `[^abc]` : Correspond à tout caractère sauf a, b ou c.

## Ancres :

- `^` : Correspond au début d'une ligne ou d'une chaîne.
- `$` : Correspond à la fin d'une ligne ou d'une chaîne.

## Quantificateurs :

- `{n}` : Correspond à exactement n occurrences du caractère précédent.
- `{n,}` : Correspond à au moins n occurrences du caractère précédent.
- `{n,m}` : Correspond à entre n et m occurrences du caractère précédent.
- `*?`, `+?`, `??`, `{n,m}?` : Version non gourmande des quantificateurs.

## Caractères d'échappement :

- `\` : Permet d'échapper un métacaractère pour le traiter comme un caractère littéral. Par exemple, `\.`, `\\`, `\\`, `\\`.

## Parenthèses :

- `(abc)` : Crée un groupe de capture pour extraire une partie correspondante de la chaîne.

## Caractères spéciaux :

- `\d` : Correspond à un chiffre (équivalent à `[0-9]`).
- `\D` : Correspond à tout sauf un chiffre (équivalent à `[^0-9]`).
- `\w` : Correspond à un caractère alphanumérique (lettres, chiffres, soulignement).
- `\W` : Correspond à tout sauf un caractère alphanumérique.
- `\s` : Correspond à un espace blanc (espaces, tabulations, sauts de ligne).
- `\S` : Correspond à tout sauf un espace blanc.

## Exemples d'utilisation :

- `^\d{3}-\d{2}-\d{4}$` : Correspond à un numéro de sécurité sociale américain.
- `[A-Za-z]+` : Correspond à un mot composé uniquement de lettres.
- `\bword\b` : Correspond à la chaîne "word" en tant que mot entier.

- `\d{1,3}(\,\d{3})*` : Correspond à un nombre avec des séparateurs de milliers (par ex., 1,000 ou 123,456,789).

## Modificateurs :

- `i` : Recherche insensible à la casse (par ex., `/motif/i` correspondra à "Motif" ou "motif").
- `g` : Correspondance globale pour trouver toutes les occurrences (par ex., `/motif/g`).

# [Regex] Exemples

## Introduction

Cette page donne des exemples de regex utiles à copier dans vos scripts ou votre code.

# . [Regex]\*

## Exemples

### Adresse email

```
^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$
```

### Numéro de téléphone:

```
^(0[1-9](?:\d{2}){4}|0[1-9](?:\d{8}|\d{2}-\d{2}-\d{2}-\d{2}))$
```

### URL

```
^(http|https|ftp)://[^\s/$.?\#].[\^\s]*$
```

### Date

- Format AAAA-MM-JJ :

```
^\d{4}-\d{2}-\d{2}$
```

- Format JJ-MM-AAAA

```
^(0[1-9]|[12][0-9]|3[01])-(0[1-9]|1[0-2])-\d{4}$
```

- Format MM-JJ-AAAA

```
^(0[1-9]|1[0-2])-(0[1-9]|[12][0-9]|3[01])-\d{4}$
```

## Adresses IP

- IPv4 :

```
^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\. (25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\. (25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\. (25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$
```

- IPv6

```
^([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}$
```

## Mot de passe fort

Il doit contenir au moins une lettre majuscule, une lettre minuscule, un chiffre et un caractère spécial, et doit faire au moins 8 caractères de long) :

```
^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$
```

## Identifiant d'utilisateur

Il doit comporter des lettres minuscules, des chiffres et des soulignements, et être de 3 à 16 caractères :

```
^[a-z0-9_]{3,16}$
```

## Numéro de carte de crédit (Visa ou MasterCard)

```
^(4|5[1-5])\d{15}$
```