

Referência Rápida de Expressões Regulares

Padrões, quantificadores, grupos, lookaheads, flags

Padrões Básicos

Metacaracteres

- Qualquer caractere (exceto nova linha)
- ^ Início da string / linha
- \$ Fim da string / linha
- * 0 ou mais do anterior
- + 1 ou mais do anterior
- ? 0 ou 1 do anterior (opcional)
- \ Escapar metacaractere

Correspondência Literal

```
hello # matches "hello" exactly
a.c # matches "abc", "alc", "a-c", etc.
.txt # matches literal ".txt"
```

Classes de Caracteres

Expressões de Colchetes

```
[abc] # Corresponder a, b ou c
[^abc] # Corresponder a qualquer coisa exceto a, b, c
[a-z] # Letra minúscula
[A-Z] # Letra maiúscula
[0-9] # Dígito
[a-zA-Z0-9] # Alfanumérico
```

Classes Abreviadas

```
\d # Dígito [0-9]
\D # Não dígito [^0-9]
\w # Caractere de palavra [a-zA-Z0-9_]
\W # Não caractere de palavra
\s # Espaço em branco [\t\n\r\f]
\S # Não espaço em branco
```

Quantificadores

Quantificadores Gananciosos

```
* # 0 ou mais (ganancioso)
+ # 1 ou mais (ganancioso)
? # 0 ou 1 (ganancioso)
{n} # Exatamente n vezes
{n,} # n ou mais vezes
{n,m} # Entre n e m vezes
```

Quantificadores Preguiçosos

```
*? # 0 ou mais (preguiçoso / não ganancioso)
+? # 1 ou mais (preguiçoso)
?? # 0 ou 1 (preguiçoso)
{n,m}? # Entre n e m (preguiçoso)
```

Quantificadores preguiçosos correspondem ao menor número de caracteres possível

Ganancioso vs Preguiçoso

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

Âncoras

```
^ # Início da string (ou linha com flag m)
$ # Fim da string (ou linha com flag m)
\b # Limite de palavra
\B # Não limite de palavra
\A # Início da string (não afetado por m)
\Z # Fim da string (não afetado por m)
```

Exemplos de Âncoras

```
^Hello # starts with "Hello"
world$ # ends with "world"
\bword\b # "word" as whole word
\Bword\B # "word" inside another word
```

Grupos e Alternância

Grupos de Captura

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

Tipos de Grupo

```
(pattern) # Grupo de captura
(?:pattern) # Grupo não capturante
(?P<name>pat) # Grupo nomeado (Python)
(?<name>pat) # Grupo nomeado (JS, .NET)
\1 \2 # Referência retroativa ao grupo 1, 2
a|b # Alternância: a ou b
```

Lookahead e Lookbehind

```
(?=pattern) # Lookahead positivo
(?!pattern) # Lookahead negativo
(?<=pattern) # Lookbehind positivo
(?<!pattern) # Lookbehind negativo
```

Exemplos de Lookaround

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

Lookarounds correspondem a uma posição sem consumir caracteres

Padrões Comuns

```
\d{1,3}(\.\d{1,3}){3} # Endereço IPv4 (básico)
[\w.+-]+@[lw-]+\.[lw.]+ # E-mail (básico)
https?://[lw./\-?&#]=]+ # URL (básico)
(?:\d{3}\w)?[-.\s]?\d{3}[-.\s]?\d{4} # Número de telefone dos EUA
\d{4}-\d{2}-\d{2} # Data (AAAA-MM-DD)
#[0-9a-fA-F]{6} # Código de cor hexadecimal
```

Estes são padrões simplificados; uso em produção pode exigir validação mais rigorosa

Flags

```
g # Global: encontrar todas as correspondências, não apenas a primeira
i # Correspondência insensível a maiúsculas
m # Multilinha: ^ / $ correspondem a limites de linha
s # Dotall: . também corresponde à nova linha
x # Verbose: ignorar espaços em branco, permitir comentários
u # Unicode: suporte completo a Unicode
```

Uso de Flags por Linguagem

```
/pattern/gi # JavaScript
re.compile(r"pat", re.I | re.M) # Python
grep -iE "pattern" # grep (extended)
```