

Sequência de caracteres que formam um padrão de pesquisa.

São usadas em validação de dados, procura e substituição, e parsing.

Quando se valida, pretende-se, normalmente, que toda a string coincida com o padrão.

Quando se procura, pretende-se que uma substring coincida com o padrão.

Existem 12 caracteres especiais que têm significados especiais nas expressões regulares: `\`, `^`, `$`, `.`, `|`, `?`, `*`, `+`, `(`, `)`, `[` e `{`. Para encontrar um destes símbolos numa string, utiliza-se uma `\` antes (`\\`, `\^`, etc).

Classe de caracteres, ou conjunto, coincide com apenas um de vários caracteres (exemplo: `gr[ae]y`: gray ou grey).

Pode-se utilizar hífen (`-`) para especificar intervalos numa classe de caracteres (exemplo: `[0-9a-fA-F]`: 0 a 9 ou a a f ou A a F).

Para negar uma classe usa-se um acento circunflexo (`^`) no início da classe (exemplo: `[^A-Za-f]`: qualquer letra exceto A a Z e a a f).

Dentro de uma classe de caracteres, os únicos caracteres especiais são `]`, `\`, `^` e `-`.

Um ponto (`.`) coincide com qualquer carácter exceto mudanças de linha.

Âncoras podem ser usadas para especificar a posição da string coincidente:

- `^` coincide a posição antes do primeiro carácter da string.
- `$` coincide logo após o último carácter da string.
- Podem ser usados os dois para validar uma string completa.

O metacaracter `\b` é uma âncora que coincide com uma posição chamada de "word boundary". Permite fazer pesquisas de palavras completas (exemplo: `\bis\b`: is).

Uma `|` permite coincidir com apenas uma de várias expressões regulares (exemplo: `cat|dog`).

O `?` faz com que o carácter precedente seja opcional (exemplo: `colou?r`: color ou colour).

Uma `*` permite que o elemento anterior se repita 0 ou mais vezes. Um `+` permite que este se repita 1 ou mais vezes.

Usando `{` e `}` podemos especificar o máximo e mínimo de repetições. Exemplos:

- `[0-9]{9}`: repete-se 9 vezes.
- `[0-9]{1, 3}`: repete-se entre 1 e 3 vezes.
- `[0-9]{2, }`: repete-se pelo menos 2 vezes.
- `[0-9]{, 3}`: repete-se no máximo 3 vezes.

Para tornar as repetições lazy, adiciona-se um `?` após o operador de repetição. Isto força o processador a realizar retrocessos mais vezes. Uma alternativa seria negar classes.

Colocando parte de um padrão entre `()` cria um grupo, que permite aplicar quantificadores e alterações a partes específicas de um padrão.

Grupos são capturados e numerados automaticamente, o que permite extrair diferentes partes de uma expressão.

Para criar um grupo que não queremos que seja capturado, começamos o grupo com `?:`

Backreferences são usadas para coincidir o mesmo texto duas vezes (`\n` ou `$n`).

Usando `?!` podemos coincidir algo que não é seguido de outra coisa.

`?<=` permite verificar se o texto precedente coincide (exemplo: `(?<= is)land:` match é land de island).

Em HTML, os elementos input têm um atributo `pattern` que obriga a coincidir com a expressão regular.

Em PHP, os padrões devem estar delimitados por `/`, `#` ou `~`. A função `preg_match()` permite utilizar expressões regulares e encontrar um resultado. A função `preg_match_all()` encontra todos os resultados. A função `preg_replace()` substitui os resultados por uma string.

Em JavaScript têm que ser delimitados por `/`. A função `test`, testa se existe um resultado. A função `match()` executa e procura uma expressão regular numa string. A função `search()` retorna o índice do primeiro resultado, se existir. A função `replace()` permite substituir um resultado.