

REG EXP

Thursday, 1 July 2021

10:44 AM

Simple strings match themselves

- `/abc/` matches `"abc"`
- `/abc/` matches `"xxabcxx"`
- `/abc/` does not match `"abxxc"`
- `/abc/` does not match `"ab"`
- `/abc/` does not match `"ABC"`

`.` matches any single character

- `/a.c/` matches `"abc"`
- `/a.c/` matches `"axc"`
- `/a.c/` does not match `"ac"`

`\` removes special meaning

- `/a\.c/` matches `"a.c"`
- `/a\\c/` matches `"a\c"`
- `/a\/c/` matches `"a/c"`

^ and \$ match string begin, end

- `/^abc/` matches `"abcd"`
- `/^abc/` does not match `"dabc"`
- `/abc$/` does not match `"abcd"`
- `/abc$/` matches `"dabc"`

[] matches any character in set

- `/a[xyz]c/` matches `"axc"`
- `/a[xyz]c/` does not match `"axyzc"`
- `/a[a-z]c/` matches `"abc"`

- `/a[a-zA-Z]c/` matches `"aBc"`
- `/a[^a-z]c/` does not match `"abc"`

QUANTIFIERS:

*** matches zero or more repeats**

- `/ab*c/` matches `"abc"`
- `/ab*c/` matches `"abbbbbbc"`
- `/ab*c/` matches `"ac"`

+ matches one or more repeats

- `/ab+c/` matches `"abc"`
- `/ab+c/` matches `"abbbbbbc"`
- `/ab+c/` does not match `"ac"`

? matches zero or one repeats

- `/ab?c/` matches `"abc"`
- `/ab?c/` does not match `"abbbbbc"`
- `/ab?c/` matches `"ac"`

{ } specifies a number of repeats

- `{n}` matches exactly n repeats
- `/ab{3}c/` matches `"abbbc"`
- `/ab{3}c/` does not match `"abbbbbc"`

{ } specifies a number of repeats

- `{n,}` matches n or more repeats
- `/ab{3,}c/` matches `"abbbc"`
- `/ab{3,}c/` matches `"abbbbbc"`

{ } specifies a number of repeats

- $\{n, m\}$ matches between n and m repeats
- `/ab{3,4}c/` matches "abbbc"
- `/ab{3,4}c/` does not match "abbbbbc"

NOTE: THE QUANTIFIERS apply to the immediate p
An item is usually a single character, a period or a c

Grouping a character:

() repeats multiple items

- `/ab+c/` matches "abbbbbc"
- `/(ab)+c/` matches "abababababc"

preceding item in the pattern.
char class.

Greediness

- `/<.+>/` matches all of "`<i>italic text</i>`"
- `/<[^>]+>/` matches only "`<i>`italic text`</i>`"

QUIZ:

Question 1 of 1

The regular expression pattern `/ab+c/` is equivalent to `/ab{1,}c/`.

👉 You are correct!

TRUE

