Use Parentheses for Grouping and Capturing

By placing part of a regular expression inside round brackets or parentheses, you can group that part of the regular expression together. This allows you to apply a quantifier to the entire group or to restrict alternation to part of the regex.

Only parentheses can be used for grouping. Square brackets define a character class, and curly braces are used by a quantifier with specific limits.

Parentheses Create Numbered Capturing Groups

Besides grouping part of a regular expression together, parentheses also create a numbered capturing group. It stores the part of the string matched by the part of the regular expression inside the parentheses.

The regex `Set(Value)?` matches `Set` or `SetValue`. In the first case, the first (and only) capturing group remains empty. In the second case, the first capturing group matches `Value`.

Non-Capturing Groups

If you do not need the group to capture its match, you can optimize this regular expression into `Set(?:Value)?`. The question mark and the colon after the opening parenthesis are the syntax that creates a non-capturing group. The question mark after the opening parenthesis is unrelated to the question mark at the end of the regex. The final question mark is the quantifier that makes the previous token optional. This quantifier cannot appear after an opening parenthesis, because there is nothing to be made optional at the start of a group. Therefore, there is no ambiguity between the question mark as an operator to make a token optional and the question mark as part of the syntax for non-capturing groups, even though this may be confusing at first. There are other kinds of groups that use the `(?` syntax in combination with other characters than the colon that are explained later in this tutorial.

`color=(?:red|green|blue)` is another regex with a non-capturing group. This regex has no quantifiers.

Regex flavors that support named capture often have an option to turn all unnamed groups into non-capturing groups.

Using Text Matched By Capturing Groups

Capturing groups make it easy to extract part of the regex match. You can reuse the text inside the regular expression via a backreference. Backreferences can also be used in replacement strings. Please check the replacement text tutorial for details.

Make a Donation

Did this website just save you a trip to the bookstore? Please make a donation to support this site, and you'll get a **lifetime of advertisement-free access** to this site!