

Regex Escape Characters

Backslash in /regex/ Expression

When you are trying to create a literal backslash match in Ruby, if you use the `/regex/` expression, the backslash parsing are only done once. Meaning that `\\` will become `\` string literal.

For example: `\\/h/` will be matching the TEXT `"\h"`

So in the context of `/regex/` expression, you just need to escape it once

Backslash in Regexp.new Expression

However, if you are creating the regular expression via `Regexp.new(expression)`, then the string you passed in will be parsed twice, that means `\\` will become `\`, then become `\` again then when used it basically means you didn't escape it at all.

The effect of `\\` and `\` is the same if used in `Regexp.new(expression)` because of the twice parsing.

If you want to actually match a string literal of `"\"` then you would have to input `\\\"` which becomes `\\` on first parse then `\` on second parse for matching string literal `\`.

Revision #2

Created 9 November 2022 17:38:52 by Tamarine

Updated 10 November 2022 14:25:55 by Tamarine