# **Regex**

Reguliari ekspresija

## “Regular expression” yra simboliu seka kuri specifikuoja paieškos modelį. Dažniausiai šis modelis yra naudojamas, kada mes vykdom paiešką string duomenų tipuose, ir ilguose tekstuose.

## Regex ekspresija aprašoma įdentan modelį tarp dviejų paverstų brūkšnių. / /. Regex ekspresija galima priskirti kintamajam.

| const regex = / / |
| --- |

| const regex = /Nemo/gi |
| --- |

## Regex galime kurti ir su new baziniu zodziu, nes RegExp Javascript yra objektas.

| let re = new RegExp('ab+c'); |
| --- |

Pvz.

| console.log(/abc/.test("abcde")); *// → true* console.log(/abc/.test("abxde")); *// → false* |
| --- |

Flags

Regex ekspresijoje galima nurodyti parametrus angl. flag. kurios keičia regex veikimo modelį.

**g:** naudoja tą patį modelį globaliai, visiems atvejams

const regex = /abc/g

**i**: padaro modeli NOT case sensitive (NEpriklausoma nuo mažųjų/didžiųjų raidžių tikrina visus atitikmenis, nepriklausomai ar tai mažoji ar didžioji raidė)

const regex = /abc/i

galima flag parametrus grupuotis:

| const regex = /abc/gi  [A-Z][a-z][0-9] |
| --- |

Populiarios regex kalbos išraiškos angl. k.

| /abc/ | A sequence of characters |
| --- | --- |
| /[abc]/ | Any character from a set of characters |
| /[^abc]/ | Any character *not* in a set of characters |
| /[0-9]/ | Any character in a range of characters |
| /x+/ | One or more occurrences of the pattern x |
| /x+?/ | One or more occurrences, nongreedy |
| /x\*/ | Zero or more occurrences |
| /x?/ | Zero or one occurrence |
| /x{2,4}/ | Two to four occurrences |
| /(abc)/ | A group |
| /a|b|c/ | Any one of several patterns |
| /\d/ | Any digit character |
| /\w/ | An alphanumeric character (“word character”) |
| /\s/ | Any whitespace character |
| /./ | Any character except newlines |
| /\b/ | A word boundary |
| /^/ | Start of input |
| /$/ | End of input |