

Tartalomjegyzék

- 1 Regular expressions
 - ◆ 1.1 Character classes
 - ◆ 1.2 Recurrence
 - ◆ 1.3 Choice
 - ◆ 1.4 Grouping
- 2 Tasks
 - ◆ 2.1 Advanced tasks

Regular expressions

Regular expressions are used to find complex patterns in text, or if we want to substitute these patterns for something else. We will use this site <https://regex101.com/#python>

- Special characters: These don't symbolize themselves, to find them in text we have to escape them with \ for example: \\$, ^ etc.

. ^ \$ * + ? { } [] \ | ()

Character classes

For the time being we only use one character patterns.

- \d: arbitrary number, \D: arbitrary character that is not a number.
- \w: arbitrary alphanumeric character, character, number, or underline (_), \W: arbitrary non-alphanumeric character.
- \s: whitespace, which is tab, end of line, space, \S arbitrary non-whitespace character.
- We can create custom character classes: [xyz], or we can make exclusions, e.g. [^xyz]. The former matches x, y or z, the latter matches any character that is not x, y or z. Using a dash we can specify intervals, e.g. [a-z] this matches all lower case characters, but for example [A-Za-z0-9] matches all uppercase, lowercase and numeric characters.
- ^: beginning of line, \$, end of line.
- A . matches any character.

Recurrence

Notation	Recurrence number	Example
*	0,1, or however many	\d* matches '123', and it even matches the empty string, <i>as well</i>
+	at least 1	\d+ matches any number of numeric characters
?	0 or 1	the?an matches 'then' and 'than' as well
{m,n}	At least <i>m</i> , at most <i>n</i> number of something, both of them are optional	:D{4,10} does not match 'DDDDDDDDDDDDDDDD'

Choice

- The pattern `alelilolu` matches any vowel. Try the `GetValue|Get|Set|SetValue` expression. What do we get for the text `SetValue`?

Grouping

We can specify groups within the expression. The following example matches any string that repeats once:

```
(.*)\1
```

We can match for HTML tags:

```
<([A-Z][A-Z0-9]*)\b[^>]*>.*/\1>
```

We can specify multiple groups, the sequence of the opening parenthesis specifies the number. Replace the ending of email addresses to `.hu!`

```
(\w+)@((\w+)\.)+(\w+)
```

Tasks

- date formats: `yyyy.mm.dd`
- Mobile numbers starting with `+36 20`, `+36 30`, `+36 70`
- Link tags (`<a>anything here`)
- Webpage addresses
- Find the BME logo with patterns on: <http://www.bme.hu/?language=en>
- 2 digit numbers divisible by 4
- leap years

Advanced tasks

- Roman numerals written with capital letters

Millenium: `M{0, 4}`, century: `CM|CD|D?C{0, 3}`, decade: `XC|XL|L?X{0, 3}`, year: `IX|IV|V?I{0, 3}`.

- Positive integers, really long numbers might contain spaces when grouped by 3 digits (1 000, 435 000 000).
- Decimal color code in HTML (3 or 6 hexa number)