

# Regulární výrazy (Regex)

**Regex** je posloupnost znaků, která tvoří vyhledávací vzor. Umožňuje definovat velmi komplexní pravidla – například „najdi všechna slova, která začínají velkým písmenem, obsahují aspoň jedno číslo a končí tečkou“.

## 1. Základní stavební kameny (Metaznaky)

Metaznaky jsou speciální symboly, které nemají doslovný význam, ale určují logiku hledání.

| Znak | Význam                                     | Příklad                              |
|------|--------------------------------------------|--------------------------------------|
| .    | Jakýkoliv jeden znak (kromě nového řádku). | a.c najde „abc“, „a1c“, „a!c“        |
| ^    | Začátek řádku.                             | ^Ahoj najde „Ahoj“ pouze na začátku  |
| \$   | Konec řádku.                               | konec\$ najde „konec“ pouze na konci |
| [ ]  | Množina znaků (výčet).                     | [aeiou] najde jakoukoli samohlásku   |
| [^ ] | Negativní množina (vše kromě).             | [^0-9] najde cokoli, co není číslice |
| \    | Escapování (zrušení speciálního významu).  | \. najde skutečnou tečku             |

## 2. Kvantifikátory (Určení počtu)

Kvantifikátory určují, kolikrát se má předchozí znak nebo skupina opakovat.

- ``*``: Nula nebo vícekrát (např. `lo*l` najde „ll“, „lol“, „loool“).
- ``+``: Jednou nebo vícekrát (např. `lo+l` najde „lol“, „loool“, ale NE „ll“).
- ``?``: Nula nebo jednou (nepovinný znak).
- ``{n}``: Přesně n-krát.
- ``{n,m}``: n až m-krát.

## 3. Třídy znaků a zkratky

Místo vypisování [0-9] existují předdefinované zkratky:

- `\d`: Jakákoli číslice (digit).
- `\w`: Jakýkoli „slovní“ znak (písmena, čísla, podtržítko).
- `\s`: Bílý znak (mezera, tabulátor, nový řádek).
- `\b`: Hranice slova (začátek nebo konec slova).

## 4. Seskupování a logické OR

- `( )`: Seskupování znaků (např. `(abc)+` hledá opakování celé sekvence „abcabc“).
- `|`: Logické „nebo“ (např. `kočka|pes` najde buď jedno, nebo druhé).

## 5. Praktické příklady ze světa IT

### Validace e-mailu (zjednodušená)

```
^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$
```

- Rozbor: Začni textem (písmena/čísla), následuje zavináč, pak doména a končí tečkou s příponou (min. 2 znaky).

### Vyhledání IP adresy

```
\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b
```

- Rozbor: Čtyři skupiny jedné až tří číslic oddělené tečkami.

### Úprava v příkazu sed

Smazání všech komentářů v konfiguračním souboru:

- `sed -i '/^#/d' config.conf`

(Najdi řádky začínající ^ mřížkou # a smaž je d.)

**Zajímavost:** Existuje slavný citát Jamieho Zawinského: „Někteří lidé, když se setkají s problémem, si pomyslí: 'Já vím, použiji regulární výrazy.' Teď mají problémy dva.“ To odkazuje na fakt, že špatně napsaný regex může být velmi nepřehledný a těžko se v něm hledají chyby.

[Zpět na Linux](#)

From:

<http://serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:

<http://serviceit.cz/doku.php?id=regex>

Last update: **2025/12/31 14:35**

