

[redacted]

[redacted] disponují
[redacted]

[redacted]

[redacted] se
[redacted] níže.

[redacted] znaky
[redacted] se
[redacted]

[redacted]
můžeme sekvenci znaků uzavřít do závorek ((a)) a pokud za pravou kulatou závorku
doplníme kvantifikátor, bude se počet opakování vztahovat na celou sekvenci znaků
uzavřenou do závorek.

- Pokud chceme dát na výběr několik variant textu (třeba Petr nebo Pavel), jako oddělovač variant použijeme metaznak | (výraz bude tedy Petr|Pavel)
- Pokud chceme přikázat, že hledaný textový řetězec se musí nacházet na začátku nebo konci prohledávaného textu, použijeme metaznaky, které nazýváme **hranice** (boundaries) nebo **ukotvení** (anchors) – na různé typy hranic se podíváme níže.

Následující tabulky zobrazují seznam kvantifikátorů a hranic.

Kvantifikátory

Kvantifikátor	Počet opakování
?	minimálně 0krát, maximálně 1krát

Použití všech zmíněných metaznaků nejlépe pochopíte na několika příkladech.

Regulární výraz	Odpovídá...
<code>a+</code>	sekvence písmen <code>a</code> (1 a více znaků)
<code>a*</code>	sekvence písmen <code>a</code> (0 a více znaků)
<code>o?kov</code>	<code>okov</code> či <code>kov</code>
<code>tel(efon)?</code>	<code>tel</code> či <code>telefon</code>
<code>telef(on ax)</code>	<code>telefon</code> či <code>telefax</code>
<code>[0-9] [1-9][0-9]</code>	čísla 0 až 99
<code>\d{2}</code>	sekvence dvou číslic desítkové soustavy (<code>00</code> , <code>01</code> , ..., <code>98</code> , <code>99</code>)
<code>[0-9a-fA-F] [1-9a-fA-F][0-9a-fA-F]+</code>	hexadecimální čísla
<code>(19 20)\d{2}</code>	letopočty 1900-2099
<code>\d{2,6}</code>	sekvence dvou až šesti číslic
<code>[^ ,.]+</code>	neprázdná sekvence znaků mezi nimiž nesmí být mezera (<code> </code>), čárka (<code>,</code>) či tečka (<code>.</code>)
<code>^P.*</code>	řetězec, který začíná písmenem <code>P</code> za nímž následuje libovolný (i nulový) počet libovolných znaků
<code>\d+0\$</code>	řetězec, který končí znakem <code>0</code> (nula), kterému předchází minimálně jedna číslice
<code>a+b</code>	<code>ab</code> , <code>aab</code> , <code>aaab</code> atd.
<code>a\b+</code>	<code>a+b</code>

Poznámky závěrem

- Popsané konstrukce odpovídají regulárním výrazům vycházejícím z Perlu – budou tedy fungovat především v Perlu, .NETu, PHP (při použití Perl-compatible regular expressions funkcí) a Javascriptu.
- V tomto úvodním článku nebyly zmíněny pokročilejší konstrukce regulárních výrazů jako zpětné odkazy (backreferences), modifikátory (modifiers), pokročilejší hranice, komentáře, tvrzení (assertions) nebo podmíněné subvýrazy (conditional subexpressions). Některé pokročilejší konstrukce jsou závislé na konkrétní implementaci regulárních výrazů v daném programovacím jazyce, proto jsou popsány ve zvláštních tutoriálech (seriálech článků).