

Znaky a výrazy

SPECIÁLNÍ ZNAKY	mají zvláštní význam, interpretem je shell
\$	dolar - prompt (výzva); vyjadřuje připravenost shellu k přijetí příkazu (výchozí označení pro běžného uživatele)
#	křížek 1) prompt (výzva); vyjadřuje připravenost shellu k přijetí příkazu (výchozí označení pro uživatele root) 2) komentář v konfiguračních souborech a skriptech; každé slovo uvedené po tomto znaku je systémem ignorováno, výjimkou je „#!“ (shebang) na začátku skriptu následovaný absolutní cestou k shellu nebo programu určenému pro spuštění souboru
/	lomítko 1) představuje kořenový adresář 2) odděluje adresáře v absolutní cestě
.	tečka; představuje aktuální (pracovní) adresář
..	dvě tečky; představují nadřazený adresář
-	spojovník 1) uvozuje volby příkazu \$ ls -a 2) čte data ze STDIN \$ gzip -cd soubor.tar.gz tar -xf -
--	dvojitý spojovník 1) uvozuje dlouhé volby příkazu \$ ls --all 2) označuje ukončení voleb příkazu (následující řetězec je považován za argument) \$ mkdir -- -dir \$ mkdir ./-dir (vytvoří adresář „-dir“)
\	zpětné (obrácené) lomítko (na konci řádku); označuje zalomení řádku příkazu
>	znaménko „větší než“ - \$PS2 (na začátku řádku); označuje pokračování řádku příkazu
Kolony:	posloupnosti příkazů oddělených jedním z následujících řídicích operátorů:
	roura; propojí standardní výstup jednoho příkazu pomocí roury se standardním vstupem jiného příkazu \$ ls -l grep "^d" wc -l (vypíše počet podadresářů v pracovním adresáři)
& 2>&1	roura a ampersand; propojí standardní výstup a standardní chybový výstup jednoho příkazu pomocí roury se standardním vstupem jiného příkazu
Seznamy:	posloupnosti příkazů nebo kolon oddělených jedním z následujících řídicích operátorů:
;	středník; označuje konec příkazu (totéž co nový řádek) \$ cd /tmp; ls -la
&	ampersand; spustí příkaz na pozadí ve vnořeném shellu (vhodné zejména pro větší úlohy; návratový kód je vždy nulový)
&&	dvojitý ampersand; spustí následující příkaz, jen pokud předchozí skončí úspěšně (vrátí nulový návratový kód) \$ mount /mnt/fd && cp -R /mnt/fd /tmp/fd && umount /mnt/fd (připojí disketu, zkopíruje její obsah do „/tmp/fd“ a disketu odpojí)
	dvojitá roura; spustí následující příkaz, jen pokud předchozí skončí neúspěšně (vrátí nenulový návratový kód) \$ [\$? -eq 0] echo \$? (vypíše návratový kód předchozího příkazu, skončil-li neúspěšně)

SPECIÁLNÍ ZNAKY	mají zvláštní význam, interpretem je shell
Složené příkazy:	skupiny příkazů, se kterými se zachází jako s jednou jednotkou
(<seznam>)	kulaté závorky; seskupují příkazy (příkazy se provedou ve vnořeném shellu) \$ (a; b) & c & (příkazy „a“ a „b“ se provedou na pozadí sekvenčně, tzn. jeden za druhým, příkaz „c“ paralelně)
{ <seznam>; }	složené závorky; seskupují příkazy (příkazy se provedou v aktuálním shellu) \$ { a; b;} & c & (příkazy „a“ a „b“ se provedou na pozadí sekvenčně, tzn. jeden za druhým, příkaz „c“ paralelně) \$ cd ~/.ssh 2> /dev/null { mkdir ~/.ssh; cd ~/.ssh; } (vstoupí do daného adresáře, pokud neexistuje, napřed ho vytvoří)
PARAMETRY	představují entity, v nichž jsou uloženy určité hodnoty
Poziční parametry:	číselné parametry uvnitř shelového skriptu představující argumenty předané skriptu na příkazovém řádku v odpovídajícím pořadí
<n> {<n>}	číselný parametr začínající od 1
Zvláštní parametry:	parametry, jejichž obsah je pouze pro čtení
<parametr> {<parametr>}	# uvnitř skriptu představuje celkový počet argumentů předaných skriptu na příkazovém řádku, * či @ uvnitř skriptu představuje všechny argumenty předané skriptu na příkazovém řádku (dochází-li k expanzi uvnitř dvojitých uvozovek, "\$*" expanduje do jednoho slova s každým pozičním parametrem odděleným mezerou, zatímco "\$@" expanduje každý poziční parametr do samostatného slova), uvnitř skriptu představuje jméno shelového skriptu, na příkazovém řádku představuje jméno současného shellu, \$ představuje PID současného shellu, ? představuje návratový kód posledního procesu spuštěného na popředí, ! představuje PID posledního procesu spuštěného na pozadí, _ představuje poslední argument posledního provedeného příkazu, - představuje aktuální volby shellu
Proměnné:	„úložiště“ informací, skládají se z názvu a přiřazené hodnoty
<promenna>=<hodnota>	definice lokální proměnné (zůstane definovaná v shellu až do jeho ukončení) \$ xy="den" (definuje proměnnou „xy“ a přiřadí jí hodnotu „den“) \$ pracovni_adresar=\$(pwd) (přiřadí proměnné hodnotu výstupu příkazu) \$ PATH="\$PATH:~/skripty" (přidá adresář „skripty“ do proměnné „PATH“) \$ PS1="[\u@\h \W]\\$ " (nastaví prompt uživatele; proměnnou „PS1“ lze definovat následovně: \u = uživatel, \h = jméno počítače, \W = pracovní adresář)
Pole:	datové struktury, které umožňují uložit více hodnot do jedné proměnné
<indexovane_pole>=(<hodnota1> <hodnota2> <hodnota3> ...)	definice indexovaného pole (uspořádaného seznamu položek); každá hodnota v poli je spojena s číselným indexem, který začíná 0 pro první položku a zvyšuje se o 1 pro každou další položku v poli, a pomocí odpovídajícího indexu se k dané hodnotě přistupuje; indexovaná pole mohou být před definováním volitelně deklarována \$ barvy=("modra" "cervena" "svetle zelena") \$ declare -a barvy=("modra" "cervena" "svetle zelena") (definuje indexované pole „barvy“ a přiřadí mu tři různé hodnoty) \$ barvy+=("zluta" "bila") (přidá do indexovaného pole další hodnoty) \$ barvy[3]="cerna" (přiřadí hodnotu „cerna“ položce indexovaného pole s indexem 3)

SPECIÁLNÍ ZNAKY	mají zvláštní význam, interpretem je shell
<pre><asociativni_pole>=(<klic1> <hodnota1> <klic2> <hodnota2> <klic3> <hodnota3> ...) <asociativni_pole>=([<klic1>]=<hodnota1> [<klic2>]=<hodnota2> [<klic3>]=<hodnota3> ...)</pre>	<p>definice asociativního pole (slovník – páry klíč-hodnota); každá hodnota v poli je spojena s jedinečným klíčem a pomocí odpovídajícího klíče se k dané hodnotě přistupuje; asociativní pole musí být před definováním explicitně deklarována</p> <pre>\$ declare -A hostname_to_ip=(["server1"]="192.168.1.1" ["server2"]="192.168.1.2" ["server3"]="192.168.1.3")</pre> <p>(deklaruje a definuje asociativní pole „hostname_to_ip“ a přiřadí mu tři různé klíče se svými hodnotami)</p> <pre>\$ hostname_to_ip["server4"]="192.168.1.4" \$ hostname_to_ip+=(["server4"]="192.168.1.4")</pre> <p>(přidá do asociativního pole další pár klíč-hodnota)</p>
EXPANZE	nahrazují určité řetězce nebo výrazy jejich odpovídajícími hodnotami nebo výsledky
Expanze složených závorek:	proces, který vytváří kombinace řetězců nebo sekvencí na základě zadaného vzoru
<pre>{<retezec>[,<retezec>]}</pre>	<p>složené závorky; definují řetězec (řetězce oddělené čárkou), jenž má daný výraz obsahovat; interval po sobě jdoucích čísel nebo písmen je vyjádřen „..“; ke každému řetězci může být přidána předpona či přípona, která se stává součástí výrazu</p> <pre>\$ mkdir p{an,lo,ru}t (vytvoří adresáře „pant“, „plot“ a „prut“)</pre> <pre>\$ mv text{1,2,3,4,5}.txt text_0{1,2,3,4,5}.txt (přejmenuje soubory „textX.txt“ na „text_0X.txt“)</pre> <pre># mv /etc/pam.d/sshd{, .broken} (přejmenuje soubor „/etc/pam.d/sshd“ na „/etc/pam.d/sshd.broken“)</pre> <pre>\$ touch .{a,b,c,d,e} (vytvoří skryté soubory daných jmen)</pre> <pre>\$ touch {1..100} (vytvoří sto souborů)</pre> <pre>\$ echo {A..Z} {a..z} {0..9} (zobrazí interval daných znaků)</pre> <pre>\$ echo {A..Z}{a..z}{0..9} (zobrazí trojmístné kombinace znaků v daném pořadí)</pre>
Vlnková expanze:	proces, který nahradí znak vlnovky odpovídající cestou k domovskému adresáři uživatele
<pre>~</pre>	<p>vlnovka; představuje domovský adresář přihlášeného uživatele (\$HOME), ~<uzivatel> domovský adresář daného uživatele, ~+ pracovní adresář (\$PWD), ~- předchozí pracovní adresář (\$OLDPWD)</p> <pre>\$ cp ~/soubor.txt /tmp</pre> <p>(zkopíruje soubor „soubor.txt“ z domovského adresáře uživatele do adresáře „/tmp“)</p>
Expanze parametrů:	proces, který nahrazuje parametry jejich hodnotami
<pre>\$<n> \${<n>}</pre>	<p>hodnota pozičního parametru; je-li expandován poziční parametr sestávající z více než jedné číslice, musí být uzavřen ve složených závorkách</p> <pre>\$./zaloza.sh /media/disk-1</pre> <p>(určí zařízení, kam se záloha provede - ve skriptu je uvedeno např. „target_dir=\"\$1“)</p>
<pre>\$<parametr> \${<parametr>}</pre>	<p>hodnota zvláštního parametru</p> <pre>\$ echo \$?</pre> <p>(vypíše návratový kód posledního spuštěného příkazu)</p>

SPECIÁLNÍ ZNAKY	mají zvláštní význam, interpretem je shell
<code>\${<promenna> \${<promenna>}}</code>	hodnota proměnné; za proměnnou musí být mezera, jinak musí být uzavřena do složených závorek (kromě případů, kdy následuje „“, „\$“, „!“ či „[]“) \$ echo \$xy (vypíše hodnotu proměnné „xy“ - „den“) \$ echo je novy ty\$xy \$ echo "je novy ty\$xy" (vypíše „je novy tyden“) \$ echo "ty\${xy}ni dovolena" (vypíše „tydenni dovolena“) \$ echo \$SHELL (vypíše přihlašovací shell uživatele)
<code>\${#<promenna>}</code>	délka hodnoty proměnné (počet znaků)
<code>\${<indexovane_pole>[n]}</code>	hodnota položky indexovaného pole; indexované pole musí být uzavřeno ve složených závorkách \$ echo "Prvni barva: \${barvy[0]}" (vypíše „Prvni barva: modra“) \$ echo \${barvy[@]} (vypíše všechny položky v indexovaném poli oddělené mezerou) \$ for barva in "\${barvy[@]}"; do echo "Barva: \$barva"; done (vypíše každou položku v indexovaném poli na samostatný řádek)
<code>\${#<indexovane_pole>[@]}</code>	délka hodnot indexovaného pole (počet položek)
<code>\${<asociativni_pole>[<klic>]}</code>	hodnota klíče asociativního pole; asociativní pole musí být uzavřeno ve složených závorkách \$ echo \${hostname_to_ip["server2"]} (vypíše „192.168.1.2“) \$ echo \${hostname_to_ip[@]} (vypíše všechny hodnoty v asociativním poli oddělené mezerou) \$ echo \${!hostname_to_ip[@]} (vypíše všechny klíče v asociativním poli oddělené mezerou) \$ for server in "\${!hostname_to_ip[@]}"; do echo "\$server: \${hostname_to_ip[\${server}]}"; done (vypíše každý klíč se svou hodnotou v asociativním poli na samostatný řádek)
<code>\${#<asociativni_pole>[@]}</code>	délka hodnot asociativního pole (počet položek)
Expanze jmen:	proces, ve kterém se používají metaznaky k zastoupení jmen existujících souborů a adresářů, které odpovídají danému vzoru
*	hvězdička; zastupuje libovolný počet jakýchkoliv znaků včetně mezery, kromě výrazů začínajících tečkou # rm -Rf /* (smaže obsah adresáře) \$ ls * ' '* (vypíše soubory a adresáře obsahující v názvu mezery)
?	otazník; zastupuje libovolný znak včetně mezery, kromě výrazů začínajících tečkou # find / -name "*.19??" (najde soubory a adresáře s příponou „.19xx“) # find / -name "???" (najde soubory a adresáře, jejichž název obsahuje alespoň tři znaky)

SPECIÁLNÍ ZNAKY	mají zvláštní význam, interpretem je shell
[...]	<p>hranaté závorky; zastupují libovolný znak v seznamu; seznam může tvořit výčet jednotlivých znaků, interval znaků (obsahuje „-“, např. A-Z, a-z či 0-9) nebo třída znaků (:[:alnum:], [:alpha:], [:ascii:], [:blank:], [:cntrl:], [:digit:], [:graph:], [:lower:], [:print:], [:punct:], [:space:], [:upper:], [:word:], [:xdigit:]); „!“ či „^“ na začátku seznamu vylučuje následující znaky (skupiny znaků)</p> <pre>\$ ls text_[ab].txt</pre> <p>(vypíše soubory „text_a.txt“ a „text_b.txt“)</p> <pre>\$ find . -name "[a-zA-Z[:digit:]]"</pre> <p>(najde soubory a adresáře obsahující v názvu pouze jedno malé či velké písmeno nebo číslici)</p> <pre>\$ rm -f .[!.*]</pre> <p>(smaže všechny skryté soubory v pracovním adresáři)</p> <pre>\$ ls ~/.ssh/id_*[!.pub]</pre> <p>(vypíše všechny soukromé ssh klíče)</p>
Aritmetická expanze:	proces, který vyhodnocuje aritmetické výrazy a nahrazuje je jejich výsledky
\$((<vyraz>)) \${<vyraz>}	<p>dolar a dvojité kulaté závorky nebo dolar a hranaté závorky; aritmetický výraz uvnitř je vyhodnocen a nahrazen svým výsledkem</p> <pre>\$ echo "2*5=\$((2*5))" \$ echo "2*5=\${2*5}"</pre> <p>(zobrazí „2*5=10“)</p> <pre>\$ ciska=(10 20 30); echo "Soucet prvnych dvou polozek = \$((ciska[0] + ciska[1]))"</pre> <p>(zobrazí „Soucet prvnych dvou polozek = 30“)</p> <pre># uid=500; for usr in a b c; do useradd -u \$uid \$usr; uid=\$((uid+1)); done</pre> <p>(vytvoří tři účty, jejichž UID začíná hodnotou 500 a narůstá o 1)</p>
Nahrazení výstupem příkazu:	proces, který nahrazuje příkaz jeho výstupem
`<prikaz>` \${<prikaz>}	<p>zpětné (obrácené) apostrofy nebo dolar a kulaté závorky; příkaz uvnitř je zpracován a nahrazen svým výstupem</p> <pre>\$ rpm -qf `which xhost` \$ rpm -qf \${which xhost}</pre> <p>(výstup příkazu „which xhost“ se předá příkazu „rpm -qf“)</p>
Nahrazení procesem:	proces, který umožňuje, aby výstup příkazu vypadal jako vstup ze souboru nebo výstup do souboru
<(<seznam>) >(<seznam>)	<p>kulaté závorky a znaménko „menší než“ nebo „větší než“; provedou seznam příkazů, jejichž výstup se uloží do speciálního dočasného souboru (pojmenované roury - FIFO), který se poté předá jako argument aktuálnímu příkazu, jenž očekává soubor, ze kterého bude číst nebo do něj bude zapisovat</p> <pre>\$ diff <(hostname) /etc/hostname</pre> <p>(porovná obsah výstupu příkazu (speciálního souboru) s obsahem daného souboru)</p> <pre>\$ cat input.txt tee >(grep a > a.out) >(grep b > b.out) >(grep c > c.out) > /dev/null</pre> <p>(filtruje data ze souboru „input.txt“ podle daných vzorů a přesměruje je do odpovídajících samostatných souborů)</p> <pre>\$ echo "\$error_message" tee >(mail -r "\$sender" -s "\$subject" "\$recipients")</pre> <p>(zobrazí chybovou zprávu a její obsah odešle e-mailem)</p>
ZNAKY PŘESMĚROVÁNÍ	přesměrují standardní vstup, výstup a chybový výstup příkazu
< 0<	<p>přesměrování standardního vstupu ze souboru (STDIN / souborový popisovač 0)</p> <pre>\$ mail tom@prompt.cz < seznam.txt</pre> <p>(napíše uživateli „tom“ email, jehož obsah se nachází v souboru „seznam.txt“)</p>

SPECIÁLNÍ ZNAKY	mají zvláštní význam, interpretem je shell
<< <oddelovac>	<p>přesměrování standardního vstupu z víceřádkového textu (here-document)</p> <pre>#!/bin/bash cat <<EOF > /etc/mujconfig.conf # Muj konfigurační soubor parametr1=hodnota1 parametr2=hodnota2 parametr3=hodnota3 EOF</pre> <p>(přesměruje více řádků textu ve skriptu až po oddělovač „EOF“ na STDIN příkazu „cat“)</p> <pre>\$ cat <<END > Jsem \$LOGNAME. <- ' > END <- '</pre> <p>(interaktivní forma)</p>
<<< <retezec>	<p>přesměrování standardního vstupu z jednoho řádku textu nebo proměnné (here-string)</p> <pre>\$ wc -w <<< "Toto je radek textu"</pre> <p>(zpočítá počet slov v řetězci)</p> <pre>\$ read -a pole_clenu <<< "\$clenove"</pre> <p>(přečte mezerami oddělený řetězec uložený v proměnné "\$clenove", rozdělí ho na jednotlivá slova a každé slovo uloží jako prvek do pole "pole_clenu")</p>
> 1>	<p>přesměrování standardního výstupu do souboru (STDOUT / souborový popisovač 1)</p> <pre>\$ ls -la > obsah_adresare.txt</pre> <p>(výstup příkazu zapíše do daného souboru, který zároveň vytvoří, pokud existuje, přepíše jeho obsah)</p> <pre>\$ cat a tee b > c</pre> <p>(výstup příkazu zapíše do souborů „b“ a „c“)</p> <pre>\$ wc -l < report > /tmp/lines</pre> <p>(počet řádků souboru „report“ zapíše do „/tmp/lines“)</p>
2>	<p>přesměrování standardního chybového výstupu do souboru (STDERR / souborový popisovač 2)</p> <pre>\$ cat soubor1 soubor2 > soubor3 2> /dev/null</pre> <p>(obsah souborů „soubor1“ a „soubor2“ zapíše do „soubor3“ a standardní chybový výstup příkazu zahodí)</p> <pre>\$ make all 2> /dev/pts/3</pre> <p>(standardní chybový výstup příkazu zobrazí v daném terminálu)</p>
2>&1	<p>přesměrování standardního chybového výstupu na standardní výstup (záleží na pořadí přesměrování STDOUT)</p> <pre>\$ ls a b > c 2>&1 \$ ls a b &> c</pre> <p>(STDOUT i STDERR zapíše pouze do souboru „c“)</p> <pre>\$ ls soubor1 soubor2 2>&1 > soubor3</pre> <p>(STDERR vypíše na obrazovku a STDOUT zapíše do „soubor3“)</p> <pre>\$ make all 2>&1 less</pre> <p>(vhodné pokud proces vypisuje více chybových hlášení)</p>
&>	<p>přesměrování standardního výstupu a standardního chybového výstupu do souboru</p> <pre># yum install -y libstdc++.x86_64 &> /dev/null</pre> <p>(veškerý výstup příkazu zahodí)</p>
>>	<p>přesměrování standardního výstupu na konec souboru</p> <pre># echo 'ALL ALL=!SUDOSUDO' >> /etc/sudoers</pre> <p>(výstup příkazu přidá na konec daného souboru, pokud soubor neexistuje, vytvoří ho)</p>
2>>	<p>přesměrování standardního chybového výstupu na konec souboru</p> <pre># fsck /dev/sda1 2>> error</pre> <p>(standardní chybový výstup příkazu přidá na konec daného souboru, pokud soubor neexistuje, vytvoří ho)</p>
ZNAMY VÝLUKY	potlačují význam speciálních znaků

SPECIÁLNÍ ZNAKY	mají zvláštní význam, interpretem je shell
\	zpětné (obrácené) lomítko; brání interpretaci následujícího znaku shellem \$ echo \\$auto (vypíše „\$auto“)
'...'	jednoduché uvozovky; brání interpretaci jakéhokoliv řetězce shellem; jednoduché uvozovky však nemohou být uzavřeny samy do sebe \$ echo '\$auto' (vypíše „\$auto“) \$ echo '\\$auto' (vypíše „\\$auto“) \$ echo '\$auto' (vypíše „jeep“)
"..."	dvojitě uvozovky; obdoba jednoduchých, avšak povolují interpretaci zástupných znaků a znaků „\$“ a „`“, „\“ má svůj řídicí význam pouze tehdy, následuje-li znak „\$“, „““, „`“, „\“ a nový řádek; dvojitě uvozovky však nemohou být uzavřeny samy do sebe \$ echo "\$auto" (vypíše „jeep“) \$ echo "\\$auto" (vypíše „\$auto“) \$ echo "\$auto" (vypíše „jeep“)

REGULÁRNÍ VÝRAZY (rozšířené)	představují vzory, které slouží k vyhledání určitých řetězců v textu, interpretem je program
Speciální znaky:	určují charakter výrazu
.	tečka; odpovídá libovolnému jednomu znaku včetně mezery
[...]	hrnaté závorky; odpovídají právě jednomu znaku v seznamu, „-“ určí interval znaků (např. A-Z, a-z nebo 0-9), „^“ na začátku seznamu vylučuje následující znaky
[[:alpha:]] / [a-zA-Z]	alfabetické výrazy
[[:lower:]]	malá písmena
[[:upper:]]	velká písmena
[[:digit:]] / [0-9]	numerické výrazy
[[:alnum:]] / [0-9a-zA-Z]	alfanumerické výrazy
[[:punct:]]	interpunkční znaménka
[[:print:]]	tisknutelné výrazy
[[:blank:]]	mezera nebo tabulátor
(...)	kulaté závorky; seskupují znaky
	roura (logické NEBO); odděluje hledané výrazy
\	zpětné lomítko; snižuje význam následujícího speciálního znaku
Poziční znaky:	určují pozici výrazu
^	stříška; odpovídá začátku řádku
\$	dolar; odpovídá konci řádku
\<...>	potlačené lomené závorky; odpovídají výrazu, který není obklopen písmenem, číslicí nebo podtržítkem
Kvantifikátory:	určují počet opakování předchozího výrazu
?	otazník; min. 0krát, max. 1krát
*	hvězdička; min. 0krát, max. neomezeně
+	plus; min. 1krát, max. neomezeně
{<n>}	číslo ve složených závorkách; právě nkrát
{<m>,<n>}	čísla ve složených závorkách oddělená čárkou; min. mkrát, max. nkrát
{<m>,>}	číslo a čárka ve složených závorkách; min. mkrát

REGULÁRNÍ VÝRAZY (rozšířené)	představují vzory, které slouží k vyhledání určitých řetězců v textu, interpretem je program
{,<n>}	čárka a číslo ve složených závorkách; max. <i>n</i> krát
Regulární výraz:	odpovídá:
a.	„a“ + jakýkoliv jeden znak
a+b	„ab“, „aab“, „aaab“ ...
a\b	„a+b“
o?hon	„ohon“ či „hon“
auto(mobil)?	„auto“ či „automobil“
mikro(fon skop)	„mikrofon“ či „mikroskop“
^(Odkud Kam):	řádek začínající řetězcem „Odkud:“ nebo „Kam:“
ha{1,3}	„ha“ či „haha“ či „hahaha“
<T[DH]>	„<TD>“ či „<TH>“
\<a.*a\>	řetězec začínající i končící písmenem „a“
\[[a-zA-Z]\]	libovolné písmeno uzavřené v „[]“
[0-9]+	alespoň jedna číslice
[0-9][1-9][0-9]	interval čísel „0“-„99“
[0-9]{2}	interval čísel „00“-„99“
(19 20)[0-9]{2}	interval čísel „1900“-„2099“
[0-9a-fA-F][1-9a-fA-F][0-9a-fA-F]+	hexadecimální čísla
^\$	prázdný řádek
[.^az\]	tečka, stříška, „a“, „z“, zpětné lomítko či spojovník
[^ ,]+	řetězec neobsahující mezeru, čárku, ani tečku
.+0\$	řádek končící znakem „0“, jemuž předchází alespoň 1 znak
^P.*(0[1-9])\$	řádek začínající znakem „P“ a končící znaky „01“-„09“
(\((\+ 00)[0-9]{1,4}\)?)?([-]?[0-9]{2,4}){2,4}([[:blank:]] \$)+	jakékoliv telefonní číslo
[a-zA-Z0-9_.-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}	jakákoliv e-mailová adresa
https?://[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}(/[^]*)?	jakákoliv webová adresa

From:
<https://www.prompt.cz/> - **Prompt.cz**

Permanent link:
<https://www.prompt.cz/znaky-a-vyrazy>

Last update: **2025/01/07 22:32**

