10 Which type of regex does $LINUX_UTIL use?

*nix util	Regex variant	Additional notes
awk	ERE	may depend on implementation
grep	BRE	grep -P switches to PCRE
egrep	ERE	N/A
less	ERE	usually ERE, the regex variant is supplied by the system
screen	plaintext	N/A
sed	BRE	Using the -E flag switches to ERE

11 More learning resources

- Regex One: an interactive tutorial for teaching regex from the ground up —> https://regexone.com/
- Regex adventure: an educational workshop —> https://github.com/workshopper/regex-adventure
- Regex Crossword: a site offering a series of games allowing you to test your regex chops using old-school brainteasers —> https://regexcrossword.com/
- Redoku: regex sudoku/puzzle —> http://padolsey.github.io/redoku/
- Regex Tuesday - Challenges: regex challenges for the daring (or the bored) —> https://callumacrae.github.io/regex-tuesday/
- Most Crazy Regexes: for the LOLs —> https://stackoverflow.com/questions/800813/what-is-the-most-difficult-challenging-regular-expression-you-have-ever-written
- Regex Humor: because regex humor is the universal language —> http://www.rexegg.com/regex-humor.html

12 Conclusion

((Coming soon))

Regular Expressions in Python and Perl

1 Introduction

((Coming soon))

2 Basic Symbols

Wat do?	How Perl do?	How Python do?
Custom character class	[...]	[...]
Negated custom character class	[^...]	[^...]
Ranges	[a-z] (with ’-’ escaped if it comes last)	[a-z] (with ’-’ escaped if it comes last)
Alternation ("or")	|	|

3 Zero-width assertions

Wat do?	How Perl do?	How Python do?
Word boundary	\b	\b
Anywhere but word boundary	\B	\B
Beginning of line/string	^ / \A	^ / \A
End of line/string	$ / \Z	$ / \Z

4 Captures and Groups

Wat do?	How Perl do?	How Python do?
Capturing group	(...) or (?<name>...)	(...) or (?P<name>...)
Non-capturing group	(?:...)	(?:...)
Backreference to a specific group	\1, \g1	\1
Named backreference	\k<name>	(?P=name)

5 Character Classes

Wat do?	How Perl do?	How Python do?
Any character (except newline)	.	.
Match a non-"word" character	\W	\W
Match a "word" character	\w or [[:word:]]	\w
Case	[[:upper:]] or [[:lower:]]	N/A
Whitespace (not including newlines)	N/A	N/A
Whitespace (including newline)	\s or [[:space:]]	\s
Match a non-whitespace character	\S	\S
Match a digit character	\d or [[:digit:]]	\d
Match a non-digit character	\D	\D
Any hexadecimal digit	[[:xdigit:]]	N/A
Any octal digit	N/A	N/A
Any graphical character excluding "word" characters	[[:punct:]]	N/A
Any alphabetical character	[[:alpha:]]	N/A
Any alphanumerical character	[[:alnum:]]	N/A
ASCII character	[[:ascii:]]	N/A

6 Lookarounds

Wat do?	How Perl do?	How Python do?
Positive lookahead	(?=...)	(?=...)
Negative lookahead	(?!...)	(?!...)
Positive lookbehind	(?<=...)	(?<=...)
Negative lookbehind	(?<!..)	(?<!..)

Lookaheads assert that the character or series of characters immediately following the current position can be represented by the given expression (here represented by '...'), while lookbehinds assert that the expression is representative of the character immediately preceeding the current position.

Positive lookarounds suggest the presence of a match, while negative lookarounds assert the absense of an expression match.

7 Multiplicity

Wat do?	How Perl do?	How Python do?
0 or 1	?	?
0 or 1, non-greedy	??	??
0 or 1, don't give back on backtrack	?+	N/A
0 or more	*	*
0 or more, non-greedy	*?	*?
0 or more, don't give back on backtrack	*+	N/A
1 or more	+	+
1 or more, non-greedy	*?	*?
1 or more, don't give back on backtrack	++	N/A
Specific number	{n} or {n,m} or{n,}	{n} or {n,m} or{n,}
Specific number, non-greedy	{n,m}? or{n,}?	{n,m}? or{n,}?
Specific number, don't give back on backtrack	{n,m}+ or{n,}+	N/A

8 Other basic regex characters

Wat do?	How Perl do?	How Python do?
Independent non-backtracking pattern	(?>...)	N/A
Anywhere but word boundary	(?i) or (?-i)	(?i) or (?-i)

9 Examples

((Coming soon))