Notation	Description	Example Regex
Symbols		
literal	Match literal string value *literal*	foo
re1|re2	Match regular expressions *re1 or re2*	foo|bar
.	Match *any character* (except \n)	b.b
^	Match *start of string*	^dear
$	Match *end of string*	/bin/*sh$
*	Match *0 or more* occurrences of preceding regex	[a-za-z0-9]*
+	Match *1 or more* occurrences of preceding regex	[a-z]+\.com
?	Match *0 or 1* occurrences of preceding regex	goo?
{N}	Match *N* occurrences of preceding regex	[0-9]{3}
{M,N}	Match from *M to N* occurrences of preceding regex	[0-9]{5,9}
[...]	Match any single character from *character class*	[aeiou]
[..x-y..]	Match any single character in the *range from x to y*	[0-9],[A-Za-z]
[^...]	*Do not match* any character from character class, including any ranges, if present	[^aeiou] [^A-Za-z0-9_]
(*|+|?|{})?	Apply "non-greedy" versions of above occurrence/ repetition symbols (*, +, ?, {})	.*?[a-z]
(...)	Match enclosed regex and save as *subgroup*	([0-9{3}])?, f(oo|u)bar
Special Characters		
\d	Match any decimal *digit*, same as [0-9] (\D is inverse of \d: do not match any numeric digit)	data\d+.txt
\w	Match any alphanumeric character, same as [A-Za-z0-9_] (\W is inverse of \w)	[A-Za-z_]\w+
\s	Match any whitespace character, same as [\n\t\r\v\f] (\S)	of\sthe
\b	Match any *word boundary* (\B is inverse of \b)	\bThe\b
\N	Match saved *subgroup N* (see (...) above)	Price: \16
\c	Match any *special character* c verbatim (i.e., without its special meaning, literal)	\., \\, *
\A (\Z)	Match *start (end) of string* (also see ^ and $ above)	\ADear
Extension Notation		
(?iLnsux)	Embed one or more special "flags" parameters within the regex itself (vs. via function/method)	(?x), (?im)
(?:...)	Signifies a group whose match is *not* saved	(?:\w\.)*
(?P<name>...)	Like a regular group match only identified with name rather than a numeric ID	(?P<data>)
(?P=name)	Matches text previously grouped by (?P<name>) in the same string	(?P<data>)
(?#...)	Specifies a comment, all contents within ignored	(?#comment)
(?=...)	Matches if ... comes next without consuming input string; called *positive lookahead assertion*	(?=.comt)
(?!...)	Matches if ... does not come next without consuming input; called *negative lookahead assertion*	(?!.net)
(?<=...)	Matches if ... comes prior without consuming input string; called *positive lookbehind assertion*	(?<=800-)
(?<!...)	Matches if ... does not come prior without consuming input; called *negative lookbehind assertion*	(?<!192\.168\.)
(?(id/name)Y|N)	Conditional match of regex Y if group with given id or name exists else N; |N is optional	(?(1)y|x