

Java J2SE "Regular Expressions" Cheat Sheet v 0.1

Metacharacters			
([{\^\$ })?*.+			
Character Classes			
[abc]	a, b, or c (simple class)		
[^abc]	Any character except a, b, or c (negation)		
[a-zA-Z]	a through z, or A through Z, inclusive (range)		
[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)		
[a-z&&[def]]	d, e, or f (intersection)		
[a-z&&[^abc]]	a through z, except for b and c: [ad-z] (subtraction)		
[a-z&&[^m-p]]	a through z, and not m through p: [a-lq-z] (subtraction)		
Predefined Character Classes			
.	Any character (may or may not match line terminators)		
\d	A digit: [0-9]		
\D	A non-digit: [^0-9]		
\s	A whitespace character: [\t\n\x0B\f\r]		
\S	A non-whitespace character: [^\s]		
\w	A word character: [a-zA-Z_0-9]		
\W	A non-word character: [^\w]		
Quantifiers			
Greedy	Reluctant	Possessive	Meaning
X?	X??	X?+	X, once or not at all
X*	X*?	X*+	X, zero or more times
X+	X+?	X++	X, one or more times
X{n}	X{n}?	X{n}+	X, exactly n times
X{n,}	X{n,}?	X{n,}+	X, at least n times
X{n,m}	X{n,m}?	X{n,m}+	X, at least n but not more than m times
Boundary Matchers			
^	The beginning of a line		
\$	The end of a line		
\b	A word boundary		
\B	A non-word boundary		
\A	The beginning of the input		
\G	The end of the previous match		
\Z	The end of the input but for the final terminator, if any		
\z	The end of the input		
Class Pattern Fields			
CANON_EQ	Enables canonical equivalence.		
CASE_INSENSITIVE	Enables case-insensitive matching.		
COMMENTS	Permits whitespace and comments in pattern.		
DOTALL	Enables dotall mode.		
MULTILINE	Enables multiline mode.		
UNICODE_CASE	Enables Unicode-aware case folding.		
UNIX_LINES	Enables Unix lines mode.		

Class Matcher Methods	
static Pattern compile(String regex)	Compiles the given regular expression into a pattern.
static Pattern compile(String regex, int flags)	Compiles the given regular expression into a pattern with the given flags.
int flags()	Returns this pattern's match flags.
Matcher matcher(CharSequence input)	Creates a matcher that will match the given input against this pattern.
static Boolean matches(String regex, CharSeq input)	Compiles the given regular expression and attempts to match the given input against it.
String pattern()	Returns the regular expression from which this pattern was compiled.
String[] split(CharSequence input)	Splits the given input sequence around matches of this pattern.
String[] split(CharSequence input, int limit)	Splits the given input sequence around matches of this pattern.
Class Matcher Methods	
Matcher appendReplacement(StringBuffer sb, String replacement)	Implements a non-terminal append-and-replace step.
StringBuffer appendTail(StringBuffer sb)	Implements a terminal append-and-replace step.
int end()	Returns the index of the last character matched, plus one.
int end(int group)	Returns the index of the last character, plus one, of the subsequence captured by the given group during the previous match operation.
boolean find()	Attempts to find the next subsequence of the input sequence that matches the pattern.
boolean find(int start)	Resets this matcher and then attempts to find the next subsequence of the input sequence that matches the pattern, starting at the specified index.
String group()	Returns the input subsequence matched by the previous match.
String group(int group)	Returns the input subsequence captured by the given group during the previous match operation.
int groupCount()	Returns the number of capturing groups in this matcher's pattern.
boolean lookingAt()	Attempts to match the input sequence, starting at the beginning, against the pattern.
boolean matches()	Attempts to match the entire input sequence against the pattern.
Pattern pattern()	Returns the pattern that is interpreted by this matcher.
String replaceAll(String replacement)	Replaces every subsequence of the input sequence that matches the pattern with the given replacement string.
String replaceFirst(String replacement)	Replaces the first subsequence of the input sequence that matches the pattern with the given replacement string.
Matcher reset()	Resets this matcher.
Matcher reset(CharSequence input)	Resets this matcher with a new input sequence.
int start()	Returns the start index of the previous match.
int start(int group)	Returns the start index of the subsequence captured by the given group during the previous match operation.