

VB.NET - GROUPING CONSTRUCTS

http://www.tutorialspoint.com/vb.net/vb.net_grouping_constructs.htm

Copyright © tutorialspoint.com

Grouping constructs delineate sub-expressions of a regular expression and capture substrings of an input string. The following table lists the grouping constructs:

Grouping construct	Description	Pattern	Matches
<i>subexpression</i>	Captures the matched subexpression and assigns it a zero-based ordinal number.	$\backslash w \backslash 1$	"ee" in "deep"
$? < name > subexpression$	Captures the matched subexpression into a named group.	$? < double > \backslash w \backslash k < double >$	"ee" in "deep"
$? < name1 - name2 > subexpression$	Defines a balancing group definition.	$((? 'Open' \backslash ([^ \backslash \backslash]) * + (? 'Close - Open' \backslash [^ \backslash \backslash]) * +) * ? (Open ? !) \$$	"(1 - 3*3 - 1)" in "3+2^(1 - 3*3 - 1)"
$? : subexpression$	Defines a noncapturing group.	Write?:Line?	"WriteLine" in "Console.WriteLine"
$? imnsx - imnsx : subexpression$	Applies or disables the specified options within <i>subexpression</i> .	$A \backslash d \{ 2 \} ? i : \backslash w + \backslash b$	"A12xl", "A12XL" in "A12xl A12XL a12xl"
$? = subexpression$	Zero-width positive lookahead assertion.	$\backslash w + ? = \backslash .$	"is", "ran", and "out" in "He is. The dog ran. The sun is out."
$? ! subexpression$	Zero-width negative lookahead assertion.	$\backslash b ? ! un \backslash w + \backslash b$	"sure", "used" in "unsure sure unity used"
$? < = subexpression$	Zero-width positive lookbehind assertion.	$? < = 19 \backslash d \{ 2 \} \backslash b$	"51", "03" in "1851 1999 1950 1905 2003"
$? < ! subexpression$	Zero-width negative lookbehind assertion.	$? < ! 19 \backslash d \{ 2 \} \backslash b$	"ends", "ender" in "end sends endure lender"
$? > subexpression$	Nonbacktracking or "greedy" subexpression.	$[13579] ? > A + B +$	"1ABB", "3ABB", and "5AB" in "1ABB 3ABBC 5AB 5AC"