

第五章 模式匹配

概念型

涉及到的特殊符号, \O[{\$*+?, 它们需要转义字符\来获得它们本身的表示。

数量词: * + ? *? {3} {2,5}

保存匹配结果, \$, \$&, \$'以及\$1, \$2 等变量。

修饰符

重要型

竖线表示至少匹配其中的一个, ex: /foo | bar | zoo/

用括号分组: ex: /(foo|bar|zoo) baggins/ or /(foo | bar)o baggins/

the m// and the s/// operators are both provide the power of double-quote interpolation, the tr/// transliteration operator does not interpolate variables; it doesn't even use regular expressions.

\$, \$&, \$' and the numbered variables are global variables implicitly localized to the enclosing dynamic scopes.

All Modifiers

Modifiers	Meaning
/i	Case insensitive
/s	Let . Match newline and ignore deprecated \$* variable
/m	Let ^ and \$ match next to embeded
/x	Ignore whitespace and permit comments in pattern
/o	Compile pattern once only

The matching operator m// . The m// operator searches the strings in the scalar EXR for PATTRER. If / or ? Is the delimiter, the initial m is optional.

m// modifiers

Modifiers	Meaning
/i	
/m	
/s	
/x	Ignore whitespace and permit comments in pattern
/o	
/g	Globally find all matches
/cg	Allow continued search after failed /g match

The s/// operator (Substitution), This operator searches a string for PATTERN, if found, replaces the matched substring with the REPLACEMENT text.

The s/// Modifiers add a /e modifier, it means evaluating the right side as an expression.

The `tr///` operation (Transliteration)

`LVALUE =~ tr/SEARCHLIST/REPLACEMENT/cds`

`tr /SEARCHLIST/REPLACEMENTLIST/cds`

This operator scans a string, character by character. And replaces each occurrence of a character found in `SEARCHLIST` with the corresponding character from `REPLACEMENTLIST`. It looks like a bit like `m//` and `s///`, though, and you can even use the `=~` or `!~` binding operators on it. `Tr//` returns the number of characters replaced or deleted. If no string is specified via the `=~` or `!~` operator, the `$_` string is altered.

The `SEARCHLIST` and `REPLACEMENT` are not variable interpolated as double-quoted strings: you may, however use those backslash sequences that map to a specific character, such as `\n` or `\015`.

Tr modifiers

<i>Modifier</i>	<i>Meaning</i>
<code>/c</code>	Complement <code>SEARCHLIST</code>
<code>/d</code>	Delete found but unreplaced characters
<code>/e</code>	Squash duplicate replaced characters

Metacharacters (元字符)

琐碎型

`\|()[]{}^$*+? \b \D \t \3 \s` some character special

`\b` matches a word boundary

`\t` matches an ordinary tab

心得型