

SIGILS
\$ - scalar
@ - array
% - hash
& - subroutine
* - glob

	ARRAYS	HASHES
whole:	@array	%hash
slice:	@array[0, 2]	@hash{'a', 'b'}
element:	\$array[0]	\$hash{'a'}

NUMBERS vs STRINGS

=	=
+	.
== !=	eq ne
< > <= >=	lt gt le ge
<=>	cmp

SCALAR VALUES

number, string, reference, glob, undef

SPECIAL VARIABLES

\$_	default variable
\$0	program name
\$/	input separator
\$\	output separator
\$	autoflush
\$!	sys/libcall error
\$@	eval error
\$\$	process ID
\$.	line number
@ARGV	command line args
@INC	include paths
@_	subroutine args
%ENV	environment

REGEX METACHARS

^	string begin
\$	str. end (before \n)
+	one or more
*	zero or more
?	zero or one
{3,7}	repeat in range
()	capture
(?:)	no capture
[]	character class
	alternation
\b	word boundary
\z	string end

REGEX MODIFIERS

/i	case insensensitive
/m	line based ^\$
/s	includes \n
/x	ignore white space
/g	global

REGEX CHARCLASSES

.	== [^\n]
\s	== [\x20\f\t\r\n]
\w	== [A-Za-z0-9_]
\d	== [0-9]
\S, \W and \D	negate

DO

use strict;
use warnings;
my \$var;
open() or die \$!; /\$userinput/
use Modules;

DON'T

"\$foo"
\$\$variable_name
'\$userinput'

LINKS

perl.com
use.perl.org
perl.apache.org
cpan.org
perldoc.com

SYNTAX

```
for(LIST){ }, for(init;cond;update){ }  
while(cond){update}, until(cond){update}  
if(cond) { } elsif (cond) { } else { }  
unless (cond) { } elsif (cond) { } else { }  
foreach (LIST)
```