

A BRIEF, SOMEWHAT RANDOM LIST OF USEFUL TERMS, COMMANDS, AND SYMBOLS

abs	gives absolute value
block	set of code in subroutines and various kinds of loops; flanked by { }
cd <name>	move into named directory/folder
chomp	removes newline/carriage return from end of a string
chop	removes single character from end of a string, regardless of what it is
close	closes file that was read from or written to
cmp	compares two strings alphabetically
exit	generally not needed at end of program, but good programming form; can also be used prematurely under certain conditions
filehandle	name given to file opened in program; by convention, all caps
flag	variable used to keep track of some state or condition (for instance, that program is currently reading a certain section of a GenBank flatfile or BLAST output)
index	gives position of first occurrence of specified substring in a string
int	returns integer part floating point number
join	combine elements of an array or some other list of scalars into a single scalar, if desired, can put a specific string between each joined element
keys	makes array of keys from a hash
last	exits loop
length	returns number of characters in a string
m	match operator; check if a variable contains specified set of characters
mkdir	create a new directory/folder
my	defines a variable as only for use in the current block, subroutine, or main program
next	goes to next iteration of a loop, skipping whatever would have followed in current iteration
open	opens file to be read from or written to (must specify FILEHANDLE and actual file name)
print	print (to command line or to output file) string in quotes following command
pop	remove last element of array (opposite of push)
push	add an element to the end of an array (opposite of pop)
rand	picks a random number in specified range
return	used in subroutine to specify value returned to main code
reverse	reverse order of elements in a string or array
s	substitution operator; s/X/Y/g substitutes X with Y throughout script (“globally”); you can also substitute a specified number of times
scalar	gives size of specified array
shift	remove first element of array (opposite of unshift)
splice	insert element at arbitrary point in an array
split	“explodes” a string into an array with each character or specified pattern of characters as a separate element
srand	sets random number seed

substr	substring operator; identifies subset of string at specified position of specified length
sort	sorts items in an array according to some criterion (usually alphabetically or numerically)
tr	translate/transliterate operator; translates a set of characters into new set of characters, individually and simultaneously; tr/ACGT/TGCA gives complement sequence
unshift	add an element to the beginning of an array (opposite of shift)
values	makes array of values from a hash
\s	all white space (space, tab, newline, formfeed, carriage return)
\t	tab
\f	formfeed
\n	newline
\r	carriage return
\$	declares variable to be scalar
@	declares variable to be array
%	declares variable to be hash
@_	special type of array used in subroutines
;	end of line
“ ”	used to specify a string that can refer to other variables inside it
' '	used to specify a string that is printed or otherwise used as it can't reference other variables
=	assignment operator (<i>not</i> equality!)
==	equality
!=	inequality
<, >	less than, greater than
eq	equality for strings
ne	inequality for strings
!	not (can also use “not”)
&&	and (can also use “and”)
	or (logical or – one, other, or both true) (can also use “or”)
xor	exclusive or (one or other true, but not both)
.	concatenate
,	separates items in a list
++	increment by 1
+=x	increment number by x
=~	binding operator; for variables with strings, says apply operation following operator to variable before operator
[]	used to specify particular element of an array (numbers start with 0, not 1)
{ }	used to specify particular key of hash
//	used to specify regular expression
<>	input operators; used to specify outside file being read
()	used to specify a list of scalars on other elements (separated by commas)

REGULAR EXPRESSION METACHARACTERS

*	look for 0 or more repetitions of character
+	look for 1 or more repetitions of character
?	look for 0 or 1 repetitions of characters
\w	look for alphanumeric character or _
\W	look for non-alphanumeric character
\s	look for whitespace character (space, tab, newline, carriage return)
\S	look for non-whitespace character
\t	look for tab
\f	look for formfeed
\n	look for newline
\r	look for carriage return
\d	look for number character
\D	look for non-number character
.	look for any character but a newline
^	look for character at beginning of string
\$	look for character at end of string
{x}	look for exactly x repetitions of character
{x,}	look for at least x repetitions of character
{x,y}	look for between x and y (inclusive) repetitions of character
	look for match to either of characters on either side (ex. a b ex. a b c)
()	provides grouping for other metacharacters
[]	specifies character class

A FEW OTHER RANDOM THINGS

use warnings	tells program to print out any warning messages that come up in running program
use strict	requires that all variables be declared before they're used; more formal programming style that can help avoid confusions/multiple uses of variables, and make script easier to debug
@ARGV	special array, predefined in perl; the different elements are arguments from command line
STDIN	“standard input;” special filehandle for user keyboard input
STDOUT	“standard output;” special filehandle that acts as default location for print statements (generally set to screen)
STDERR	“standard error;” special filehandle directs error statement to screen or designated error file

SOME PERL-RELATED WEB PAGES YOU MIGHT FIND USEFUL

<http://www.perl.com/> - the main perl page associated with O'Reilly (publishers of *Beginning Perl For Bioinformatics*); very comprehensive reference for all things perl

<http://www.perl.org/> - another general page

<http://www.cpan.org/> - “Comprehensive Perl Archive Network;” collection of modules and much other info about perl