

Perl Cheat Sheet

Functions

Get information on a function by typing, e.g., `perldoc -f chomp` at the command line.

Scalar variables	<code>while (defined (\$x=<>)) {code}</code>	False if variable has never been set (or when you try to read past the end of an input file)
	<code>length(\$x)</code>	Length of a string
	<code>chomp(\$line);</code>	Remove a newline at the end of a string
	<code>\$short=substr (\$long, 2, 5)</code>	Characters 3-7 of \$long (first char is 0!)
Arrays	<code>push @arr, \$x</code>	Add to end of array
	<code>\$x = pop @arr;</code>	Remove last element of array, put in \$x
	<code>shift @arr;</code> (See also <code>unshift</code>)	Remove first element of an array, put in \$x
	<code>\$size = scalar @arr;</code>	Number of things in the array
	See also: split, join, splice, sort	split string->array, join array->string, delete part of array, sort array in many ways
Hashes	<code>@key = keys %hash</code>	The lookup terms in the hash
	<code>if (exists \$hh{"Hsp"}) {...}</code>	See whether hash %hh has a value for key Hsp
Input/Output and Files	<code>open(HANDLE, ">outfile")</code> or <code>die "Can't open \$outfile: \$!\n"</code>	Open outfile for writing, and associate it with filehandle HANDLE. Use " <code><infile</code> " for reading
	<code>print \$x;</code> <code>print HANDLE \$x;</code>	Prints to standard output (screen), Print to filehandle HANDLE
	<code>warn "Something wrong\n";</code>	Prints to standard error (screen)
	<code>\$x=<HANDLE></code>	Read a line from filehandle HANDLE, put in \$x
	<code>close(HANDLE);</code>	Stop reading/writing a previously opened file
Exit	<code>exit;</code>	Exits the program
	<code>die "Something broke!\n";</code>	Exits the program with error message

Operators and Loops

Assign value	<code>\$x = 1</code>	Sets variable to a value. Don't confuse with <code>==</code> , which tests whether numerical values are equal
Math	<code>print 1 * (2 + 3/4)</code>	Regular math symbols
	<code>10%3==1; 12%3==0</code>	Modulus (remainder) operator
	<code>\$x += 4;</code>	Same as <code>\$x=\$x+4</code> ; Also <code>-=</code> <code>*=</code> <code>/=</code>
	<code>\$x++;</code>	Same as <code>\$x=\$x+1</code> ;
Conditions	<code>if (.1 == 0.1) {print "same num"}</code>	Are numbers equal? Don't confuse with <code>=</code> or <code>eq</code>
	<code>if (1 != 2) {print "diff num"}</code>	Are numbers different?
	<code>> < >= <=</code>	Are numbers greater than, less than, etc.
	<code>if ("a" eq "a") {print "same text"}</code>	Does text have exactly the same letters?
	<code>if ("A" ne "a") {print "diff text"}</code>	Does text have different letters?
	<code>if ((\$x > 1) && (\$x < 2)) {code}</code>	Logical AND (true if both sides are true)
	<code>if ((\$x > 10) (\$x < -10)) {code}</code>	Logical OR (true if one or both sides are true)
Loops	<code>foreach my \$i (1 .. 100) {code}</code> (for and foreach are equivalent)	Sets <code>\$i</code> to 1 and does <code>code</code> . Sets <code>\$i</code> to 2, ... up to (and including) 100
	<code>while (\$a < \$b) {code}</code>	Does <code>code</code> while the condition is true (If condition is false, never enters the loop.)

Matching and Regular Expressions

Test for Match	=~	Test for match	if (\$x =~ /abc/) { ...}	Does \$x have the string "abc" anywhere in it?
	!~	Test for non-match	if (\$x !~ /abc/) { ...}	Does \$x NOT have the string "abc" anywhere in it?
	\$_	Default variable	if (/abcd/) { s/bc/x/ }	// and s// work on \$_ by default, no =~ needed
Substitute	s//	Do a Substitution	\$x =~ s/abc/def/;	Replace (only) first occurrence of "abc" in \$x with def
Match/ Sub Options	i	Ignore case.	/abc/i	Matches abc, ABC, aBc, etc.
	g	Global substitution.	s/a/c/g	Replace ALL occurrences
Special Match Items	.	Any one character (except \n)	/a.c/	"arc", "a9c", but not "ac".
	[]	Any one of.	/[abc]/	Any one of "a", "b", or "c". [a-zA-Z] matches any letter
	\d	Digit (Same as [0-9])	/\d\d:\d\d/	"10:30" (but not "9:30")
	\s	Space, tab, or newline	/^\s*\$/	An empty line.
	\	Literally match special characters: + * () / [] \ { } ^ \$ @	/1+2/	"1+2", not "1112". The backslash "quotes" or "escapes" the plus sign.
Item Locations	^	Beginning of a line	/^a/	"arginine" but not "valine".
	\$	End of a line	/a\$/	"beta" but not "beat".
Item Repetitions	?	An optional thing	/ab?c/	"ac" or "abc".
	*	Any number of copies OR nothing at all	/a*/	"", "a", "aaa".
	+	Any number of copies	/a+b/	"ab" or "aaab" but not "b".
	{ }	m to n copies	/ab{2,4}c/	"abbc", "abbbc", "abbbbc", but not "abc" or "abbbbbc"
Misc		One or the other	/abc def/	"abc" or "def"
	()	Capture parts of match in numbered variables AND group things together for repetition, etc.	/a(b(..)e)f/ /a(bc)+d/	"abcdef". This will also set \$1 to "bcde" and \$2 to "cd". "abcd" or "abcbcbcbed"