

# TCL Quick Reference

## Basic Language Features

|       |                                                                 |
|-------|-----------------------------------------------------------------|
| #     | comment (continues to end of line)                              |
| " "   | allows embedding whitespace in arguments;<br>substitutions made |
| { }   | group arguments; substitutions not made                         |
| [ ]   | command substitution; replace with result of<br>command         |
| \$var | variable substitution                                           |
| ;     | command separator                                               |

## Backslash Substitution

|          |                 |
|----------|-----------------|
| \b       | backspace       |
| \[       | open bracket    |
| \f       | form feed       |
| \]       | close bracket   |
| \n       | newline         |
| \\$      | dollar sign     |
| \r       | carriage return |
| \<space> | space           |
| \t       | tab             |
| \;       | semi-colon      |
| \v       | vertical tab    |
| \"       | double-quote    |
| \{       | left brace      |
| \<newln> | newline         |
| \}       | right brace     |
| \\       | backslash       |
| \ddd     | octal digits    |

## Built-in Variables

env  
errorCode  
errorInfo

### Operators (in decreasing order of precedence)

|          |                                                            |
|----------|------------------------------------------------------------|
| - ~ !    | unary minus, bit-wise NOT, logical NOT                     |
| * / %    | multiply, divide, remainder                                |
| + -      | add, subtract                                              |
| << >>    | left and right shift                                       |
| <> <= >= | boolean comparisons                                        |
| == !=    | boolean equal, not equal                                   |
| &        | bit-wise AND                                               |
| ^        | bit-wise exclusive OR                                      |
|          | bit-wise inclusive OR    &&                    logical AND |
|          | logical OR                                                 |
| x?y:z    | conditional operator                                       |

All operators support integers.

All support floating point except ~, %, <<, >>, &, ^, and |

Boolean operators can also be used on strings.

### Regular Expressions

|               |                                       |
|---------------|---------------------------------------|
| regex   regex | match either expression               |
| regex*        | match zero or more of regex           |
| regex+        | match one or more of regex            |
| regex?        | match zero or one of regex            |
| .             | any single character (except newline) |
| ^             | match beginning of line               |
| \$            | match end of line                     |
| \c            | match character c                     |
| c             | match character c                     |

|       |                                      |
|-------|--------------------------------------|
| []    | match set of characters              |
| [a-z] | match range of characters            |
| [^]   | match characters not in range or set |
| ()    | group expressions                    |

### Keywords

append *varName value [value value ...]*  
array anymore *arrayName searchId*  
array donesearch *arrayName searchId*  
array names *arrayName*  
array nextelement *arrayName searchId*  
array size *arrayName*  
array startsearch *arrayName*  
break  
case *string [in] patList body [patList body ...]*  
case string *[in] {patList body [patList body ...]}*  
catch *command [varName]*  
cd [*dirName*]  
close *fileId*  
concat *arg [arg ...]*  
continue  
error *message [info] [code]*  
eof *fileId*  
error *\$errMsg \$savedInfo*  
eval *arg [arg ...]*  
exec *arg [arg ...]*  
exit [*returnCode*]

expr *arg*  
file atime *name*  
file dirname *name*  
file executable *name*  
file exists *name*  
file extension *name*  
file isdirectory *name*  
file isfile *name*  
file lstat *name varName*  
file mtime *name*  
file owned *name*  
file readable *name*  
file readlink *name*  
file rootname *name*  
file size *name*  
file stat *name varName*  
file tail *name*  
file type *name*  
file writable *name*  
flush *fileId*  
for *start test next body*  
foreach *varname list body*  
format *formatString [arg arg ...]*  
gets *fileId [varName]*  
glob [-nocomplain] *filename [filename ...]*  
global *varname [varname ...]*  
history  
history add *command [exec]*  
history change *newValue [event]*  
history event [*event*]  
history info [*count*]  
history keep *count*  
history nextid  
history redo [*event*]  
history substitute *old new [event]*  
history words *selector [event]*  
if *test [then] trueBody [else] [falseBody]*  
incr *varName [increment]*  
info args *procname*

info body *procname*  
info cmdcount  
info commands [*pattern*]  
info default *procname arg varname*  
into variable *varname*  
info exists *varName*  
info globals [*pattern*]  
info level [*number*]  
info library  
info locals [*pattern*]  
info procs [*pattern*]  
info script  
info tclversion  
info vars [*pattern*]  
join *list* [*joinString*]  
lappend *varName value* [*value value ...*]  
lindex *list index*  
linsert *list index element* [*element element ...*]  
list *arg* [*arg ...*]  
llength *list*  
lrange *list first last*  
lreplace *list first last* [*element element ...*]  
lsearch *list pattern*  
lsort *list*  
open *fileName* [access]  
proc *name args body*  
puts *fileld string* [newline]  
pwd  
read *fileld*  
read *fileld* newline  
read *fileld numBytes*  
regexp [-indices] [-nocase] *exp string* [*matchVar*] [*subMatchVar subMatchVar ...*]  
regsub [-all] [-nocase] *exp string subSpec varName*  
rename *oldName newName*  
return [*value*]  
scan *string format varname1* [*varname2 ...*]  
seek *fileld offset* [*origin*]  
set *varname* [*value*]  
source *fileName*

split *string* [*splitChars*]  
string compare *string1 string2*  
string first *string1 string2*  
string index *string* charIndex  
string last *string1 string2*  
string length *string*  
string match *pattern string*  
string range *string first last*  
string tolower *string*  
string toupper *string*  
string trim *string* [*chars*]  
string trimleft *string* [*chars*]  
string trimright *string* [*chars*]  
tell *fileId*  
time *command* [*count*]  
trace variable *name ops command*  
trace vdelete *name ops command*  
trace vinfo *name*  
unknown *cmdName* [*arg arg ...*]  
unset *name* [*name name ...*]  
uplevel [*level*] *command* [*command ...*]  
upvar [*level*] *otherVar myVar* [*otherVar myVar ...*]  
while *test body*

Jeff Tranter 12-Aug-1992

Based on TCL version 6.4