

CONSTRUCTS

*list* is any sequence of commands separated by **;** or **&**, which are always interchangeable.

**if** *list*; **then** *list*  
**if** *list*; **then** *list* **]** ...  
**else** *list* **]**

*name* **[ in word ... ]**  
*list*  
*name* **in word ... ; { list }**  
**each** *name* (*word ...* )

LOBBING  
so options **GLOB**, **EXTENDED\_GLOB**, **KSH\_GLOB**, **NULL\_GLOB**, **NOMATCH**, **SH\_GLOB** **GLOB\_DOTS**. *x*, *y*, ... are any pattern. **#** and **##** require grouping various characters; those and **~**, **^** require **EXTENDED\_GLOB**.

*list*  
**end**  
**while** *list*; **do** *list*; **done**  
**until** *list*; **do** *list*; **done**  
**repeat** *word*; **do** *list*; **done**  
**repeat** *word sublist*  
**case** *word in*  
  [ *pattern* ) *list* ;; ] ...  
**esac**  
**case** *word* { [ *pattern* ) *list* ;; ] ... }  
**select** *name* [ **in word ...** ]; **do** *list*; **done**

**Subshell:** ( *list* )  
**Current shell:** { *list* }  
**function** *word* [ ( ) ] ... { *list* }  
  *word ...* () { *list* }  
  *word ...* () *sublist*  
    **time** [ *pipeline* ]  
**Condition:** [[ *exp* ]]  
Other constructs depend on the options  
**NO\_SHORT\_LOOPS** and **CSH\_JUNKIE\_LOOPS**  
and should be avoided in scripts.

Any string  
Any character  
Any of the enclosed characters  
  **]** Character classes where *X* may be:  
    Alphanumeric,  
    Alphabetic,  
    Space or tab,  
    Control character,  
    Decimal digit,  
    Printable non-whitespace character,  
    Lowercase character,  
    Printable character,  
    Printable, not alnum or space,  
    Whitespace character,  
    Uppercase character,  
    Hexadecimal digit.  
Above use locales, may be combined with other characters e.g. **[--[:xdigit:]]**  
  **]** Any character except those enclosed

**<x-y>** Any number between *x* and *y* inclusive; both optional, defaults 0, ∞  
**~X** Anything not matching *X*  
(*X|Y*) Either *X* or *Y*  
*X~Y* Pattern *X*, but not *Y*  
(*X|Y~Z*) Either *X* or (*Y* but not *Z*)  
**X#** Zero or more occurrences of *X*  
**X##** One or more occurrences of *X*  
(*X*) Grouping of (part of) pattern.  
**\*\*/** (As path segment) short for (**\*/**): match all subdirectories  
**\*\*\*/** The same, following symbolic links  
Globbing flags appear in the form (**#X**) and require the **EXTENDED\_GLOB** option. They may appear in groups. *X* may be:

**I** Case sensitive: cancel **i** and **I**  
**b** Activate backreferences for parentheses. **\$match**, **\$mbegin**, **\$mend** arrays give **match** string, beginning/end indices  
**B** Deactivate backreferences, negating **b**  
**m** Set **\$MATCH**, **\$MBEGIN**, **\$MEND** for string  
**M** Deactivate **m**.  
**num** Allow *num* errors in matches (0 to turn off)  
**s** Match only at start of string (use in *param expn*)  
**e** Match only at end of string  
Globbing modifiers appear in parentheses after a pattern (usually *and*'ed):  
  / directory  
  . plain file  
  @ symbolic link  
  = socket  
  p named pipe (**FIFO**)

executable plain file (0100)			
device file (character or block)			
block special			
character special			
readable (0400)			
writable (0200)			
executable (0200)			
group-readable (0040)			
group-writable (0020)			
group-executable (0010)			
world-readable (0200)			
world-writable (0200)			
world-executable (0200)			
setuid (04000)			
setgid (02000)			
files with the sticky bit (01000)			
chmod-like access permissions			
e.g. <code>FTO?</code> or <code>F:u+w,go-w:</code>			
eval <code>str</code> , use file ( <code>\$REPLY</code> ) if status 0			

<code>ddev</code>	or set <code>\$reply</code> to file array
<code>l[- +]<i>ct</i></code>	on device number <i>ddev</i>
<code>l</code>	link count <i>ct</i> or less (+) or more (-) than <i>ct</i>
<code>u</code>	owned by current effective uid
<code>g</code>	owned by current effective gid
<code>uid</code>	owned by uid <i>uid</i> ; may also take forms <code>.name</code> , <code>!name!</code> , ... or <code>(name)</code> , <code>{name}</code> , ...
<code>ggid</code>	owned by <i>gid</i> , as for <i>uid</i> .
<code>a[Mwln][- +]<i>n</i></code>	accessed (less than, more than) <i>n</i> days (months, weeks, hours, minutes) ago
<code>m[Mwln][- +]<i>n</i></code>	modified ditto
<code>c[Mwln][- +]<i>n</i></code>	inode changed ditto
<code>L[kKmWpP][- +]<i>n</i></code>	size in bytes (or kb, mb, blocks) = (or <, >) <i>n</i> negate following qualifiers

Options as set by default: these options appear with **no** in front in option listings; **to** turns single-letter option off (shown in parentheses)

EXPORT	Export all new shell params (-a)	BEEP†	Beep on errors etc. (+B)	CSH_NULL_GLOB	Only one glob must match
PS_LAST_PROMPT	Back to prompt after list	BG_NICE†	Lower priority of bg jobs (-6)	DVORAK	Use Dvorak keyboard for spelling
PS_TO_END	End of word after completion	BRACE_CCL	<code>foo{ab}</code> → <code>fooa foob</code>	EQUALS†	Perform <code>=cmd</code> expansion
PS_HISTORY	Append history to file	BSD_ECHO	Builtin echo works like in BSD	ERR_EXIT	Exit shell on error (-e)
CD	Directory as command does <code>cd</code> (-J)	CDABLE_VARS	<code>cd foo</code> like <code>cd ~foo</code> (-T)	EXEC†	Execute commands (+n)
LIST	List on ambiguous completion (-9)	CHASE_DOTS	Resolve links when .. in dir	EXTENDED_GLOB	Use #, ~ and ~ in patterns
MENU	Menu after second TAB	CHASE_LINKS	Resolve symlinks in directories (-w)	EXTENDED_HISTORY	Save timestamp to history file
NAME_DIRS	Params with paths become names	CHECK_JOBS†	Report job status at exit	FLOW_CONTROL†	^S, ^Q do flow control
PARAM_KEYS	Clever del after param completion	GLOBBER†	> to existing file needs >  (+C)	FUNCTION_ARGZERO†	Set \$0 on function or source
PARAM_SLASH	<code>\$path&lt;TAB&gt;</code> → <code>\$path/</code>	COMPLETE_ALIASES	Completion uses unexpanded aliases	GLOB†	Perform globbing (+F)
PUSHD	Make <code>cd</code> act like <code>pushd</code> (-N)	COMPLETE_IN_WORD	Complete at cursor point in word	GLOBAL_EXPORT†	typeset -x applies globally
REMOVE_SLASH	Strip slash after completion	CORRECT	Correct command spelling (-0)	GLOBAL_RCS†	Use /etc startup files
RESUME	cmd can behave like <code>%cmd</code> (-W)	CORRECT_ALL	Correct spelling of all args (-O)	GLOBAL_ASSIGN	scalar==* globs on right
ATTEN†	Error on bad glob pattern (+2)	CSH_JUNKIE_HISTORY	Single ! is last command	GLOB_COMPLETE	Complete globbing with menu
HIST†	Use !hist on cmd line (+K)	CSH_JUNKIE_LOOPS	Lists can be list; end	GLOB_DOTS	Leading dots match wildcards (-4)
GLOB_QUAL†	Use glob quals with just parens	CSH_JUNKIE_QUOTES	No unescaped newlines in quotes	GLOB_SUBST	Text from params can glob
AUTO_LIST†	List only on second tab	CSH_NULL_CMD	Don't use \$NULLCMD, \$READNULLCMD	HASH_CMDS†	Hash commands when run



substitute longest match of <i>pattern</i> by <i>repl</i> <i>name/pattern/repl</i> substitute shortest match <i>name//pattern/repl</i> substitute all non-overlapping longest matches <i>name/#pattern/repl</i> subst if <i>pattern</i> at start of string <i>name/%pattern/repl</i> subst if <i>pattern</i> at end of string <i>name:./pattern/repl</i> subst if <i>pattern</i> matches entire string <i>pecj</i> count length of scalar or words of array <i>pecj</i> turn on (off) <b>RC_EXPAND_PARAM</b> <i>pecj</i> turn on (off) <b>SH_WORD_SPLIT</b> <i>pecj</i> turn on (off) <b>GLOB_SUBST</b> <i>c:modj</i> apply history modifier <i>mod</i> <i>name\dotsj ... j</i> perform both sets of modifications on value I.B. does not do extra lookup, see (P)	<b>C</b> capitalise words <b>V</b> make special characters visible <b>q</b> quote result with \ <b>qq</b> quote result with ' <b>qqq</b> quote result with " <b>qqqq</b> quote result with \$' ... ' <b>q</b> remove one level of shell quoting <b>%</b> Expand prompt escapes <b>%%</b> Expand as prompt with current settings <b>X</b> Report parse errors with quotes, patterns <b>c</b> <i>\${#name}</i> counts characters <b>w</b> <i>\${#name}</i> counts words <b>w</b> As w, but count empty words <b>k</b> With assoc include keys <b>v</b> With assoc include values <b>p</b> Use print escapes in args below <b>F</b> Join words with newlines <b>f</b> Split on newlines <b>z</b> Split using ordinary parsing <b>t</b> Substituted description, not value	Flags with delimiters; use any pair of chars in place of colon, also matched <>, (), {}, [] <b>l:expr::string1::string2:</b> Pad words on left to <i>expr</i> chars using <i>string1</i> repeated (default space), <i>string2</i> appears just once <b>r:expr::string1::string2:</b> Ditto padded on right <b>j:string:</b> Join words using <i>string</i> (occurs before splitting) <b>s:string:</b> Split words at <i>string</i> Flags applying with <i>\${...#...}</i> or <i>\${...%...}</i> <b>s</b> search substrings too <b>I:expr:</b>	Search/substitute <i>expr</i> th match <b>M</b> Include matched portion <b>R</b> Include unmatched portion (Rest) <b>B</b> Include index of beginning <b>E</b> Include index of end <b>N</b> Include length of match  Summary of rules for substitution <b>1</b> Nested substitution, <i>\${\${...}}</i> <b>2</b> Subscript of parameter by name, <i>\${name[i]}</i> <b>3</b> (P) flag <b>4</b> <i>"\${...}"</i> joining <b>5</b> Nested subscript, <i>\${\${...} [i]}</i> <b>6</b> #, %, /. : modifications <b>7</b> (j) flag or space joining <b>8</b> (s), (f), (z) or = splitting <b>9</b> Shell word splitting (no flags) <b>10</b> (e) flag <b>11</b> (l) or (r) padding  Flags in indexing: usage <i>\$name[i]index</i> etc. <b>e</b> Backward compatibility only <b>w</b> Index by words of scalar <b>s:string:</b> Separate words with <i>string</i> <b>p</b> Use print escapes in following <i>s</i> <b>f</b> Index by lines: same as <i>pws:\n:</i> <b>r</b> Reverse index array/substring/word <b>R</b> For assocs, match against values <b>k</b> As <i>r</i> , but last match (all for assocs) <b>K</b> In assoc, keys are patterns; get first <b>i</b> As <i>r</i> , but return index <b>I</b> For assocs match against keys <b>n:expr:</b> As <i>I</i> , but last match (all for assocs) Use <i>expr</i> 'th first/last match <i>b:expr:</i> <i>r</i> , <i>R</i> , <i>i</i> , <i>I</i> start search at <i>expr</i> th elt.
---	---	--	---

ISTORY  
also parameters **histchars**, **HISTFILE**, **HISTSIZE**, **SAVEHIST** and options **APPEND\_HISTORY**, **CSH\_JUNKIE\_HISTORY**, **EXTENDED\_HISTORY**,  
**ALLOW\_CLOBBER**, **HIST\_IGNORE\_DUPS**, **HIST\_IGNORE\_SPACE**, **HIST\_NO\_STORE**, **HIST\_VERIFY**, **BANG\_HIST**, **HIST\_BEEP**,  
**EXPIRE\_DUPS\_FIRST**, **HIST\_FIND\_NO\_DUPS**, **HIST\_IGNORE\_ALL\_DUPS**, **HIST\_NO\_FUNCTIONS**, **HIST\_REDUCE\_BLANKS**,  
**SAVE\_NO\_DUPS**, **INC\_APPEND\_HISTORY**, **SHARE\_HISTORY**.

s:

start history substitution unless after space, newline, =, ( immediately previous command command line <i>n</i>	<i>i -n</i> <i>i str</i> <i>i ?str[?]</i> <i>i #</i>	line <i>n</i> before current last line beginning with <i>str</i> last line containing <i>str</i> current command so far	<i>i {...}</i> <i>i "</i>	insulate history reference no more expansion this line
--	---	--	------------------------------	---

s: separated from event by ':'

first word on line (command) <i>n</i> th argument of command first argument of command last argument of command	<b>%</b> <i>x-y</i> <i>-y</i> <b>*</b>	word matched by <i>?s</i> range of words same as <i>0-y</i> all arguments	<b>x*</b> <i>x-</i>	same as <i>x-\$</i> same but omit word <i>\$</i>
--	---	--	------------------------	---

ifiers: also with globbing and parameters

(head) strip last path cpt remove suffix <i>.suf</i> leave only suffix <i>suf</i> (tail) leave only last path cpt repeat last substitution don't execute new command quote words from further subst	<b>q</b> <b>x</b> <b>l</b> <b>u</b> <b>s/old/new[/]</b> <b>g</b>	remove one level of quotes same but split words at space all letters lower case all letters upper case replace <i>old</i> by <i>new</i> (string) (before <i>s</i> ) change every occurrence	<b>f</b> <b>F:expr:</b> <b>w</b> <b>w:sep:</b>	repeat till no further change same but max <i>expr</i> changes (as prefix) apply to each word same but separate words on sep
---	---	--	---	---

PARAMETERS

al parameters: arrays are lower case except **status**;  
marked† are assignable:

Last bg PID	<b>?</b> <b>pipestatus</b> <b>_</b> <b>CPUTYPE</b> <b>EGID†</b> <b>EUID†</b> <b>ERRNO</b> <b>GID†</b> <b>HOST</b>	Last prog status Array of statuses for pipeline Last arg of prev cmd CPU determined at run time Effective GID Effective UID System error no. Current GID Current host name	<b>LINENO</b> <b>LOGNAME</b> <b>MACHTYPE</b> <b>OLDPWD</b> <b>OPTARG</b> <b>OPTIND</b> <b>OSTYPE</b> <b>PPID</b> <b>PWD</b> <b>RANDOM†</b>	Input line no. User name Machine type Previous working dir. Value, index of last <b>getopts</b> option OS type PID of parent proc. Current working dir. Random integer: assign to seed.
Pos. param count Current PID Shell flags set				
Pos. params as array Same as <b>argv[0]</b>				

<b>DS</b> <sup>†</sup>	Seconds since start of shell	<b>HOME</b>	Default target for <b>cd</b> cmd.	<b>PS1</b>	Prompt used by editor
<b>IFS</b>	Incremented for each zsh	<b>IFS</b>	Word separators for input	<b>PROMPT2</b> , <b>PS2</b>	Continuation prompt
<b>LS</b>	Names of signals	<b>KEYTIMEOUT</b>	Time to waits for key in sequence	<b>PROMPT3</b> , <b>PS3</b>	Prompt used by <b>select</b> cmd.
<b>RL</b>	Name of shell terminal	<b>LANG</b>	General locale setting	<b>PROMPT4</b> , <b>PS4</b>	Execution trace prompt
<b>RL</b>	Idle time of tty (secs.) or -1	<b>LC_ALL</b>	Overrides <b>LANG</b> and other <b>LC_*</b>	<b>psvar</b> , <b>PSVAR</b> <sup>†</sup>	Replace <b>%v</b> in prompts
<b>UID</b>	UID	<b>LC_COLLATE</b>	Determines character ordering	<b>READNULLCMD</b>	Command used with only input readir.
<b>NAME</b> <sup>†</sup>	username	<b>LC_CTYPE</b>	Determines types of characters	<b>REPORTTIME</b>	Longer commands print usage (secs.)
<b>NAME</b>	Machine manufacturer	<b>LC_MESSAGES</b>	For messages; not used by zsh	<b>RPPROMPT</b>	
<b>NAME</b>	Shell invocation name	<b>LC_NUMERIC</b>	For decimal point, number separator	<b>RPST</b>	
<b>ERSION</b>	ID of zsh version	<b>LC_TIME</b>	Date and time format	<b>SAVEHIST</b>	Prompt displayed at right of line
		<b>LINES</b>	No. of lines on terminal	<b>SPROMPT</b>	Max no. of lines in history file
	parameters used by shell ( <sup>†</sup> colon-separated path)	<b>LISTMAX</b>	No. of files to list without asking	<b>STTY</b>	Prompt used for spelling correction
	Export to change <b>argv[0]</b>	<b>LOGCHECK</b>	How often to check <b>watch</b> (secs.)	<b>TERM</b>	Args. to follow <b>stty</b> ,
	Line speed (zero to ignore)	<b>MAIL</b>	File to check for mail	<b>TIMEFMT</b>	export to run before external cmd.
<b>h</b> , <b>CDPATH</b> <sup>†</sup>	Directories search for <b>cd</b> command	<b>MAILCHECK</b>	How often to check <b>MAIL</b> (secs.)	<b>TMOU</b>	Type of terminal for editing
<b>SIZE</b>	No. of columns on terminal	<b>mailpath</b> , <b>MAILPATH</b> <sup>†</sup>	List of files to check for new mail. Can follow each with <b>?message to print</b>	<b>TMPPREFIX</b>	Format of process time reports
<b>SIZE</b>	Max size of dir. stack			<b>watch</b> , <b>WATCH</b> <sup>†</sup>	SIGNALRM if idle this long (secs.)
<b>T</b>	Default editor for <b>fc</b> cmd.				Path to temp files ( <b>/tmp/zsh</b> )
<b>re</b> , <b>FIGIGNORE</b> <sup>†</sup>	Suffixes ignored for completion				List of users to watch log in/out
<b>h</b> , <b>FPATH</b> <sup>†</sup>	Path to search for autoload fns.				(also <b>all</b> , <b>notime</b> , <b>%</b> tty, <b>@</b> host)
<b>chars</b>	three chars: 1) stat of history ( <b>!</b> ), 2) quick history sub ( <b>^</b> ), 3) comment ( <b>#</b> )	<b>module_path</b> , <b>MODULE_PATH</b> <sup>†</sup>	Not used by shell, probably used by <b>man</b> cmd.	<b>WATCHFMT</b>	Format of watch reports
<b>CHARS</b>	same as <b>histchars</b>			<b>WORDCHARS</b>	Non-alphanumeric characters used
<b>FILE</b>	Where to save shell history	<b>NULLCMD</b>	Used for redirs, with no cmd.	<b>ZBEEP</b>	as part of a word by editor
<b>SIZE</b>	Max history lines internally	<b>path</b> , <b>PATH</b> <sup>†</sup>	Where to search for commands	<b>ZDOTDIR</b>	Sequence to output instead of beeping
		<b>POSTEDIT</b>	Output when line editor exits		Where to find <b>.zshrc</b> etc.
		<b>PROMPT</b> , <b>prompt</b>			

pt escape sequences: those with <sup>†</sup> can use integer count *n*, which must immediately follow **%**. Default is 1 except for **%\_**.

<b>A %%</b> <sup>†</sup>	<b>%T</b>	Time in 24 hour format	<b>%?</b>	Return status of last command
<b>A )'</b> <sup>†</sup>	<b>%*</b>	Same with seconds	<b>%_</b> <sup>†</sup>	Parser status, <i>n</i> for max level
<b>†</b>	<b>%n</b>	<b>\$USERNAME</b>	<b>%E</b>	Clear to end of line
<b>\$PWD</b>	<b>%N</b>	Name of script, sourced file, function	<b>%#</b>	<b>#</b> if root, else <b>%</b>
<b>\$PWD</b> , but use <b>~abbrevs</b>	<b>%i</b>	Line number inside <b>%N</b>	<b>%v</b> <sup>†</sup>	<i>n</i> th element of <b>\$psvar</b>
Current history event no.	<b>%w</b>	Date as <b>day--dd</b>	<b>%{...%}</b>	String which does not move cursor
The current value of <b>\$SHLV</b>	<b>%W</b>	Date as <b>mm/dd/yy</b>	<b>%&lt;string&lt; %&gt;string&gt; [%[&lt;string] % [&gt;string]</b>	Truncate string on <b>L</b> or <b>R</b> ,
Full hostname	<b>%D</b>	Date as <b>yy--mm--dd</b>		<i>n</i> gives max length.
Host up to <i>n</i> th dot	<b>%D</b>		<b>%c</b> <sup>†</sup> <b>%.</b> <sup>†</sup>	Component of <b>\$PWD</b> (deprecated)
Start standount, bold, underline	<b>%D{string}</b>	Use <b>strftime</b> to format <i>string</i>	<b>%C</b>	Same but don't expand <b>~</b> 's
Stop corresponding mode	<b>%l</b>	Current tty		
Time in 12 hour format				

for ternary expressions in prompts, format *r.true-text.false-text*, integer count *n* may be omitted or follow ‘(’. Test is true if:

- Title'd path has  $>= n$  elts
- Ditto for absolute path
- Current minute is *n*
- Current hour is *n*
- Current day of month is *n*
- Month is *n* (Jan = 0)
- Weekday is *n* (Sun = 0)

any expressions in `$WATCHFMT`, format `%(char.true-text.false-text)`, can be used with `1`, `n`, `m` or `M` (true if non-empty value for corresponding *%*), or `a` (true for login, false for logout).

### CONDITIONS

tests: followed by a file name

1

### true if file

- exists
- block special
- character special
- directory
- exists
- plain file
- as setgid bit set
- symbolic link
- as sticky bit set
- TFO/pipe
- readable
- as size  $> 0$
- as setuid bit set
- writable

- ? Last exit status was *n*
- # Running as uid *n*
- # Running as gid *n*
- g \$SHLVL  $>= n$
- L \$SECONDS  $>= n$
- S \${#psvar}  $>= n$
- v At least *n* shell constructs
- True if shell is privileged
- i

Escape sequences in `$WATCHFMT`:

- %n Name of user

- %a 'logged on' or 'logged off'
- %l User's tty
- %M Full remote host name
- %m Host to first ','
- %S %u %B Start standout, underline, boldface
- %s %u %b Stop corresponding mode
- %t %@ Time in 12-hour format
- %T Time in 24-hour format
- %w Date as day-dd
- %W Date as mm/dd/yy
- %D Date as yy-mm-dd

- x executable/dir. readable:
- L symbolic link
- O owned by UID
- G owned by GID
- S socket
- N access time not newer than mod time

Other tests with single argument:

- n string, length  $> 0$
- o option, is set
- t fd, open to tty
- z string, length zero

Two argument tests (`[[a test b]]`):

- nt file *a* newer than *b*

- ot file *a* older than *b*
- ef names refer to same file
- =
- == *string* matches *pattern*
- != ... does not match
- < ASCII before
- > ASCII after
- eq Numbers equal
- ne Numbers unequal
- lt Numeric  $a < b$
- gt Numeric  $a > b$
- le Numeric  $a \leq b$
- ge Numeric  $a \geq b$

Also grouping ( ... ), negation !, and &&, or ||; special handling of /dev/fd.