

SPECIAL CHARACTERS

& ; | * ? ' " ` [] () \$ < > { } ^ # / \ % ! ~ -

(spaces have significance)

Be careful with the use of these characters at the command line or in file names. If you have unexpected results with a command or a script it may be related to one of these.

&	- used to place jobs in the background, <i>ex: mozilla &</i>
;	- separates commands, <i>ex: ls ; cd / ; ls</i>
 	- used to “pipe” processes, <i>ex: cat filename col -b > newfile</i>
*	- wildcard, one or more characters
?	- wildcard, ONE character only
' " `	- various uses within shells and scripts (single quote, quote, and grave)
[] ()	- used to pass variables and define ranges
\$	- represents a process or a variable, <i>ex: echo \$SHELL</i>
< >	- redirection
{ }	- used to pass parameters, see find examples later in course
^	- up caret - often used to define keys, <i>ex: stty erase ^H (sets backspace)</i>
#	- pound, used at the front of a comment, ignored by shell and scripts
/	- path or escape sequences
\	- escape special meaning in command, e.g. <code>"\rm"</code> - disables the alias <code>"rm -i"</code>
%	- used by sed as a parameter, see later examples of use in vi and sed
!	- bang, used to escape a shell or vi, also combined with pound to instruct “command interpreter”, <i>ex: #!/bin/sh</i>
~	- tilde, used to indicate home directory on many systems, other uses
-	- dash, used to provide switch information to commands