

Bash Quoting

- ▶ Quoting removes the special meaning of certain characters or words to the shell
- ▶ Can be used to:
 - disable special treatment for special characters,
 - prevent reserved words from being recognized
 - prevent parameter expansion

Three kinds of Quoting:

- ▶ Backslash character: \
- ▶ Single Quotes
- ▶ Double Quotes

Backslash

- ▶ A non-quoted backslash (\) is the escape character.
- ▶ It preserves the literal value of the next character that follows
- ▶ with the exception of <newline>
- ▶ If a \`<newline>` pair appears, and the backslash is not itself quoted,
 - the \`<newline>` is treated as a line continuation (that is, it is removed from the input stream and effectively ignored).

Single Quotes

- ▶ Enclosing characters in single quotes preserves the literal value of each character within the quotes
- ▶ A single quote may not occur between single quotes, even when preceded by a backslash
- ▶ To put a single quote in a single quoted expression need three parts:
 - echo 'first part'\''third part'

Double Quotes

- ▶ Enclosing characters in double quotes preserves the literal value of all characters within the quotes with exceptions:
- ▶ Exceptions: \$, `, \, and, (when history expansion is enabled) !.

Double Quote Exceptions

1. The characters \$ and ` retain their special meaning within double quotes
2. \ retains its special meaning only when followed by
 - a. \$
 - b. `
 - c. “ (use \” within double quotes)
 - d. \
 - e. <newline>
3. If enabled, history expansion will be performed unless an ! appearing in double quotes is escaped using a backslash. The backslash preceding the ! is not removed.

Another special form

- ▶ Words of the form `'string'` are treated specially:
- ▶ `\a` alert (bell)
- ▶ `\b` backspace
- ▶ `\e` an escape character
- ▶ `\f` form feed
- ▶ `\n` new line
- ▶ `\r` carriage return
- ▶ `\t` horizontal tab
- ▶ `\v` vertical tab
- ▶ `\\` backslash
- ▶ `\'` single quote
- ▶ `\nnn` the eight-bit character whose value is the octal value `nnn` (one to three digits)
- ▶ `\xHH` the eight-bit character whose value is the hexadecimal value `HH` (one or two hex digits)
- ▶ `\cx` a control-`x` character

Examples

- ▶ echo "\$HOME"
/home/tgk (exception 1)
- ▶ echo "\\$HOME"
\$HOME (exception 2a)
- ▶ echo "\!this"
\!this (exception 3)