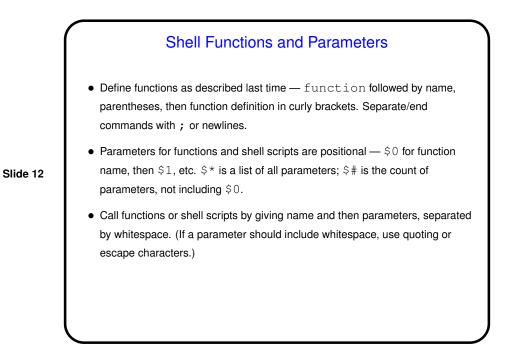
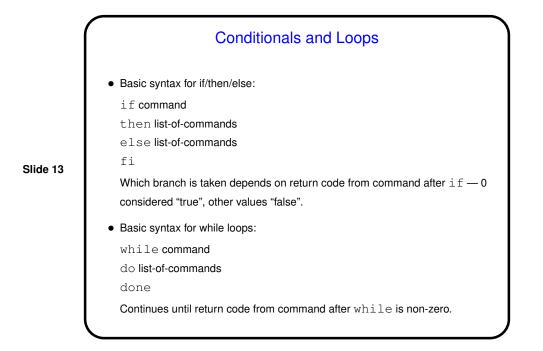


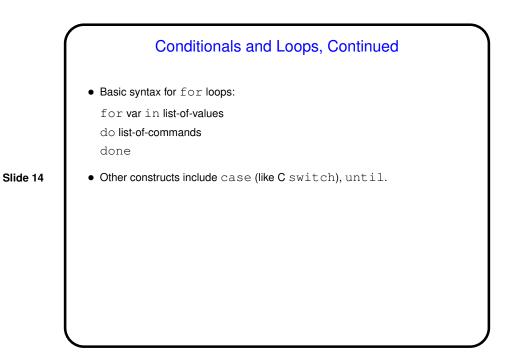
Quoting and Escape Characters
Normally bash breaks input into "words" based on whitespace, expands wildcards, performs variable substitutions (e.g., \$HOME), and a fair amount of other stuff.
When that's not what you want:

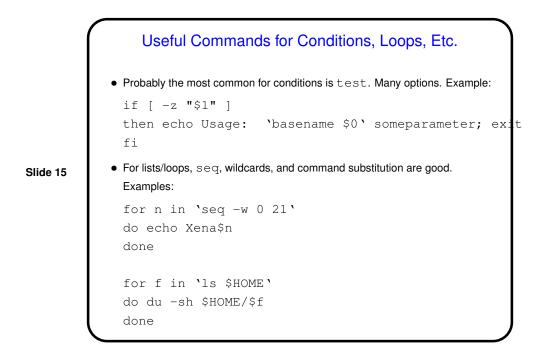
Precede "special" characters with escape character (backslash).
Use double quotes to inhibit all of the above except variable substitution.
Use single quotes to inhibit all of the above.

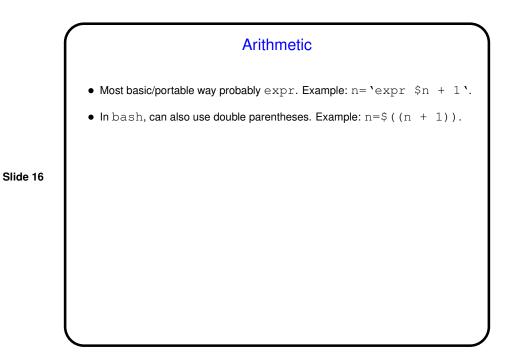
Can "inline" output of one command as parameters of another using backquotes. Example: vim `find . -name "*.c"` The "inlined" command can even be a pipeline. Example: ls -ld `echo \$PATH | sed 's/:/ /g'`









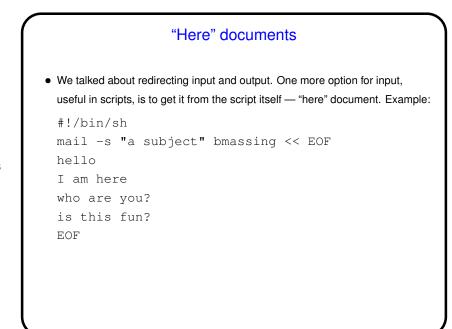


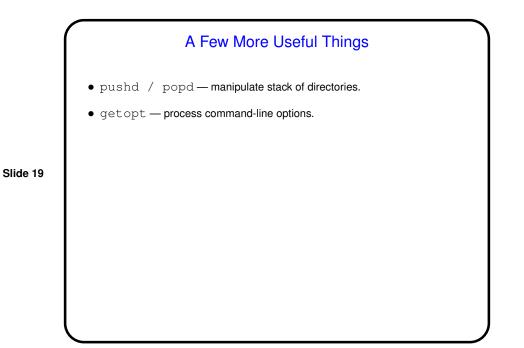
Reading from Standard Input

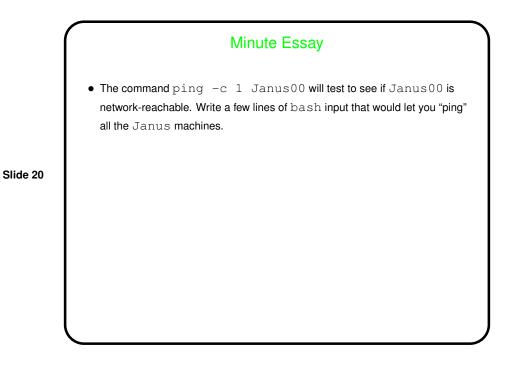
• To read from shell's / script's standard input: read. Example: echo "Do you really want to do this? (y/n)" read ans

if [".\$ans" = ".y"]

Slide 17







	Minute Essay Answer
 One p 	possible answer:
for do	n in `seq -w 0 21`
done	ping -c 1 Janus\$n e