



Shell Input as a Programming Language, with the shell as its interpreter:
Variables (usually untyped).
Expressions (arithmetic and logical).
Conditionals (if/then/else) and loops.
Functions.
I will talk about bash, but most shells provide similar functionality, just sometimes with different syntax.

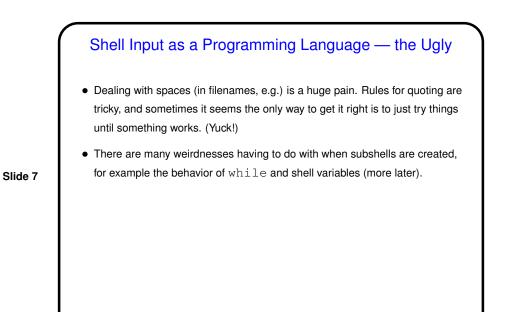
Slide 5

Shell Input as a Programming Language — the Good

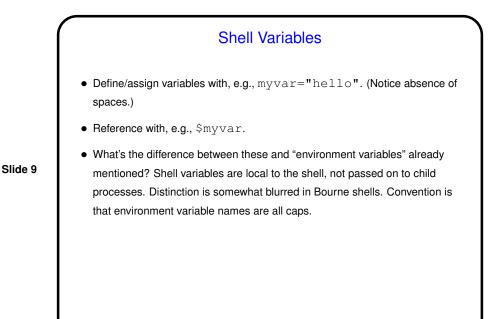
- Interactive shells are a kind of REPL (read, evaluate, print loop) for the shell's language. So you can use the various features interactively or use them to write "scripts" in the same way you can test out ideas in Scala's REPL and then use them in programs (except that the REPL is mostly useful for testing/development, whereas using shell features such as loops interactively can be useful).
- Any UNIX/Linux system will have a shell of some sort, while which "real" programming languages are available might vary.

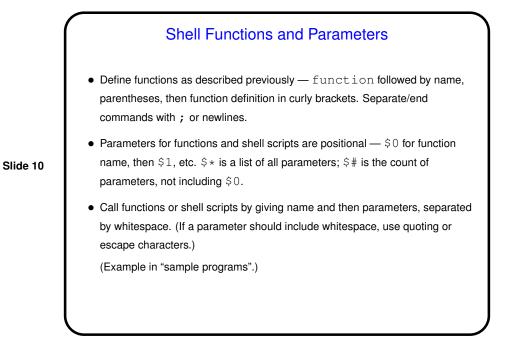
Shell Input as a Programming Language — the Bad

- Writing portable scripts is tough. Sticking to the sh subset of bash helps, as does avoiding GNU-only commands and extensions, but how to do that ... (It's a little like writing portable C.)
- What you can do is somewhat limited, and scripts of any size are apt to be ugly.
- Advice: For long and complex scripts, a scripting language such as Perl or Python may be a better choice than a shell script.

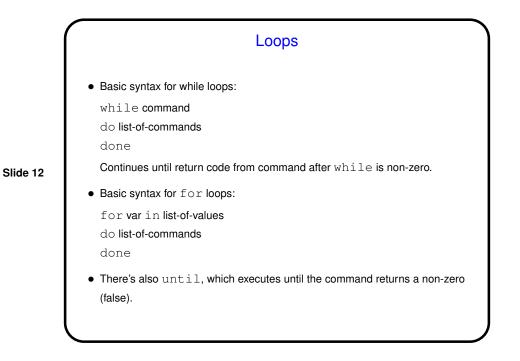


Shell Scripts
A "shell script" is just a sequence of things you could type at the shell prompt, collected in a (text) file.
Normally, first line of script is # ! followed by path for shell (/bin/bash, e.g.), and the file is marked "executable" (with chmod). But you can also execute commands in file anyfile via sh anyfile.
With the exception of the first line, lines starting with # are comments.

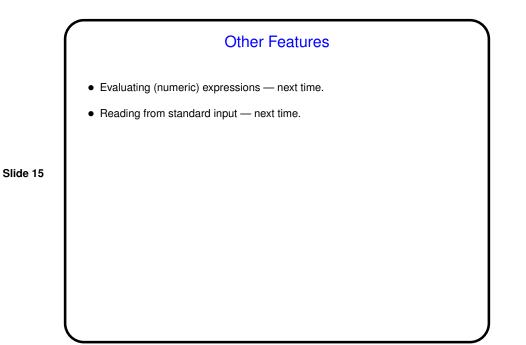


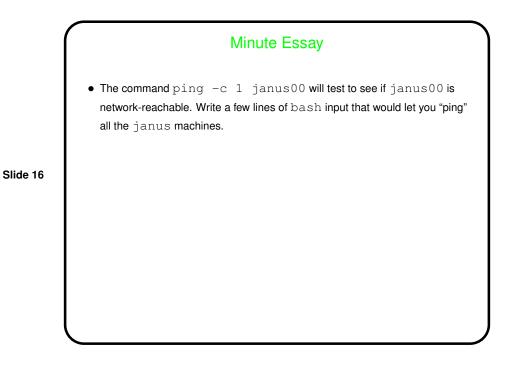


Conditionals
Basic syntax for if/then/else:
 if command
 then list-of-commands
 else list-of-commands
 fi
Slide 11
Which branch is taken depends on return code from command after if - 0
 considered "true", other values "false".
Probably the most common command test (commonly abbreviated as
 square brackets). Many options. Example:
 if [-z "\$1"]
 then echo Usage: `basename \$0` someparameter; ex:t
 fi
 case (like C switch) also available.



```
More Examples
• Rename all .htm files in the current directory to .html (-v isn't really
necessary but does show you what's being done):
   for f in 'ls *.htm'
   do
        mv -v $f 'basename .htm'.html
   done
• Descend into each of several subdirectories and launch a subshell (exit to
   move on):
   for d in d1 d2
   do
        pushd $d ; pwd ; ls ; bash ; popd
   done
```





```
Minute Essay Answer
• One possible answer:
   for n in 'seq -w 0 21'
   do
        ping -c 1 janus$n
   done
• Another answer (contributed by a student one year):
   for n in 'ruptime | grep janus | awk ' {print $1}'
   do
        ping -c 1 janus$n
   done
```