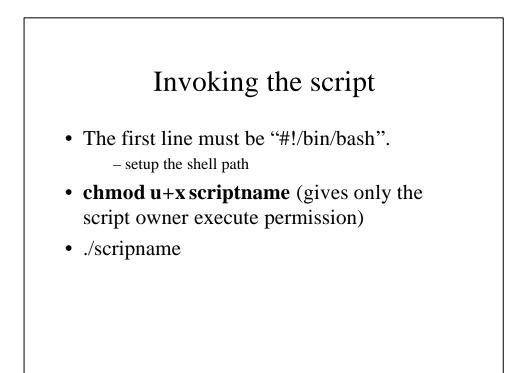
Introduction to Bash Shell

What is Shell?

- The shell is a command interpreter.
- It is the layer between the operating system kernel and the user.

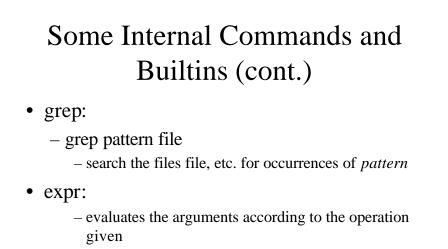
Some Special characters used in shell scripts

- #:Comments
- ~:home directory



Some Internal Commands and Builtins

- getopts:
 - parses command line arguments passed to the script.
- exit:
 - Unconditionally terminates a script
- set:
- changes the value of internal script variables.
- read:
 - Reads" the value of a variable from stdin
 - also "read" its variable value from a file redirected to stdin
- wait:
 - Stop script execution until all jobs running in background have terminated

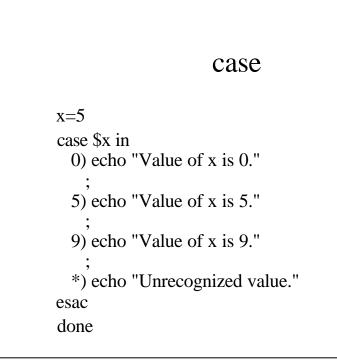


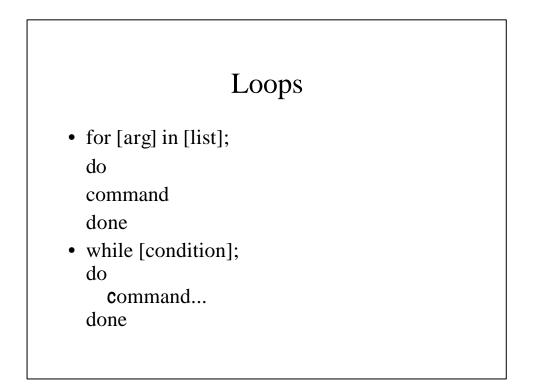
- y=`expr \$y + 1` (same as y=\$((\$y+1))

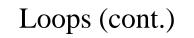
I/O Redirection

- >: Redirect stdout to a file, Creates the file if not present, otherwise overwrites it
- <: Accept input from a file.
- >>: Creates the file if not present, otherwise appends to it.
- <<:
 - Forces the input to a command to be the shell's input, which until there is a line that contains only *label*.
 - cat >> mshfile << .</p>
- |:pipe, similar to ">",

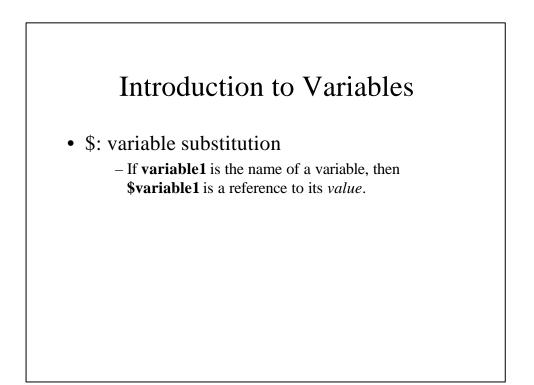
if
if [condition] then
command1
elif # Same as else if
then
command1
else
default-command
fi







- break, continue
 - break command terminates the loop
 - continue causes a jump to the next iteration of the loop



Pattern Matching

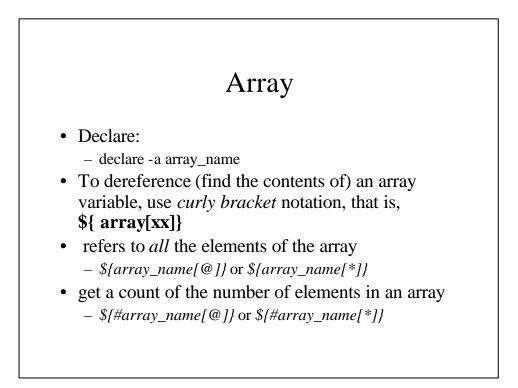
- \${variable#pattern}
- \${variable##pattern}
- \${variable%pattern}
- \${variable%%pattern}

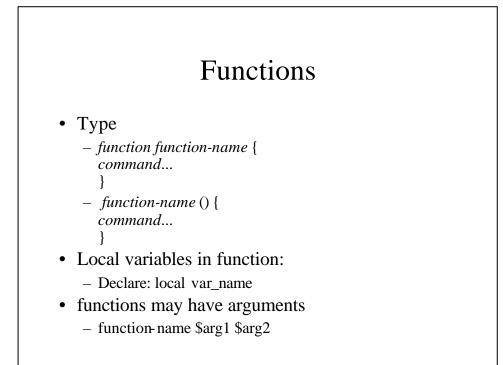
Examples of Pattern Matching

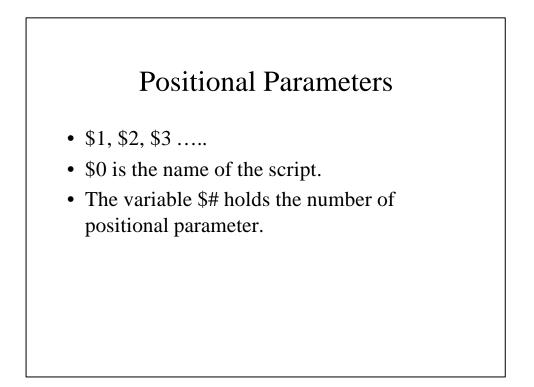
x=/home/cam/book/long.file.name echo \${x#/*/} echo \${x##/*/} echo \${x%.*} echo \${x%%.*} cam/book/long.file.name long.file.name /home/cam/book/long.file /home/cam/book/long

Aliases

- avoiding typing a long command sequence
- Ex: alias lm="ls -l | more"

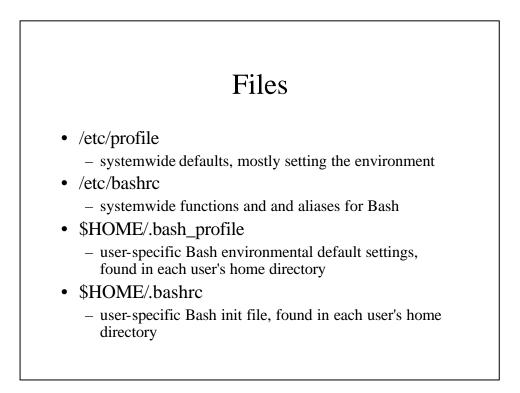






Positional Parameters in Functions

- \$1, \$2, \$3....
- Not from \$0



Debugging

- The Bash shell contains no debugger, nor even any debugging-specific commands or constructs.
- The simplest debugging aid is the output statement, echo.
- Set option
 - -n: Don't run command; check for syntax error only
 - -v: Echo commands before running them
 - -x: Echo commands after command-line processing