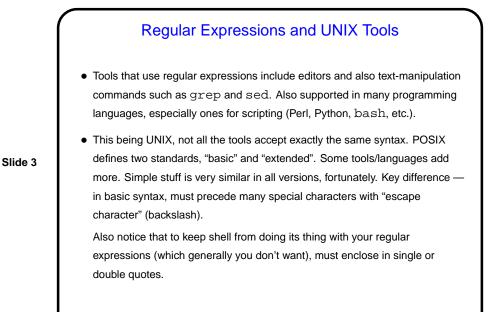


Regular Expressions From (old) Wikipedia definition: A regular expression (abbreviated as regexp, regex or regxp) is a string that describes or matches a set of strings, according to certain syntax rules. Regular expressions are used by many text editors and utilities to search and manipulate bodies of text based on certain patterns. Idea has roots in formal theory of languages, where the "languages" (sets of strings) described by regular expressions are exactly the ones accepted by finite state automata.



Character Literals and Metacharacters

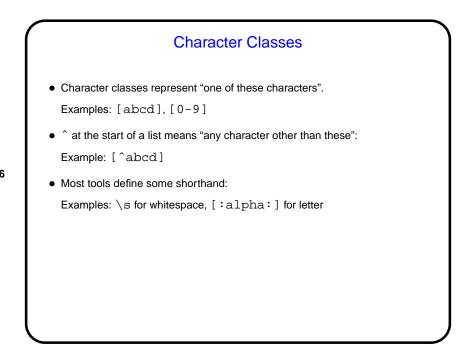
Most characters represent themselves.
hello matches what?

Other characters are "special" (metacharacters):

matches start of line
matches end of line
matches any character (except newline)

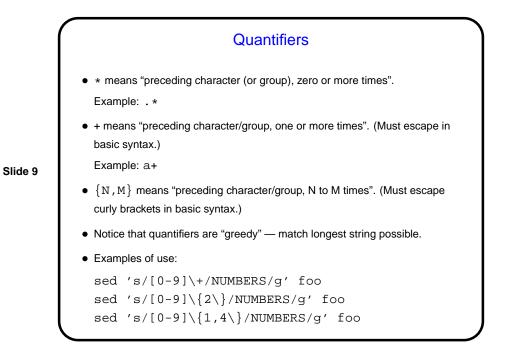
To use these as regular character literals, "escape" with a backslash.

```
Character Literals and Metacharacters, Continued
• Examples of use:
    grep "hello" foo
    grep "hello$" foo
    grep "hello$" foo
    grep "h.llo" foo
    grep "h.llo" foo
```



Exam	ples of use:
gre	o 'h[ae]llo' foo
sed	's/[A-Z]/!/g' foo
sed	's/[A-Za-z0-9]/!/g' foo
sed	's/[^A-Za-z0-9]/!/g' foo
sed	's/[[:alnum:]]/!/g' foo
sed	's/[^[:print:]]/!/g' foo

"OR" (Alternation)
UNIX pipe symbol (|) separates alternatives. (Must escape in basic syntax.) Example: cat | dog
(What about AND? Usually don't need it, or can get the same result another way. For grep, pipe one grep into another.)
Example of use: grep 'cat\|dog' foo



Grouping in Regular Expressions • Use parentheses to group. (Must escape them in basic syntax.) Example: (abc)(def) Example: (abc)* • Can then "backreference" groups, with \1, \2, etc. Example: (abc)(.*)\1 • Examples of use: sed 's/\(hello\|bye\)\+//g' foo sed 's/\((^.\)\(.*\)/\2\1/' foo

