# Regex:

**?** Represents any **one** character

**\*** Represents any **set** of characters

**[xbc]** Represents any **one** of the characters listed within the bracket

**[a-z]** Represents any character between the defined range

I/O

• Input / Output

• **STDOUT:** Standard output of command line programs

• **STDIN:** The source of input(s) for a program

• **STDERR:** Standard error output of a command line program

Redirection

**>** Redirects the standard output of a command into a file; replaces the contents of a file

**>>** Appends into the end of a file

**<** Imports the contents of a file into the command

**<<** Appends the contents of a file into the command

**2>** Redirects standard error of a command into a file

**2>>** Appends standard error of a command into the end of a file

**&>** Redirects standard error and standard output when redirecting text

**&>> •**Appends standard error and standard output when redirecting text

Pipe

Command1 | command2

xargs

Reads items from the standard input and allows commands to be run on the items.

<commands> |<xargs <command>

**Example:** ls | grep test | xargs rm –fv

»» Lists all items in the current directory, then filters the results for the string *test*, then performs a file removal with verbose output. This basically removes all files that have the string *test* in them.

Some more operators

In bash, you can run multiple commands based on the following format:

»» <Command> <option> <Command>

• **Options:**

**;** Run the following command even if the previous command fails or succeeds

**&&** Run the following command only if the previous succeeds or has no errors

**||** Run the following command only if the previous fails or results in error

**&** Run the previous command in the background

**File Tests**

• **-a <FILE> •** True if **<FILE>** exists; may cause conflicts

• **-e <FILE> •** True if **<FILE>** exists

• **-f <FILE> •** True if **<FILE>** exists and is a regular file

• **-d <FILE> •** True if **<FILE>** exists and is a directory

• **-c <FILE> •** True if **<FILE>** exists and is a character special file

• **-b <FILE> •** True if **<FILE>** exists and is a block special file

• **-p <FILE> •** True if **<FILE>** exists and is a named pipe (FIFO)

• **-S <FILE> •** True if **<FILE>** is a socket file

• **-L <FILE> •** True if **<FILE>** exists and is a symbolic link

• **-h <FILE> •** True if **<FILE>** exists and is a symbolic link

• **-g <FILE> •** True if **<FILE>** exists and has sgid bit set

• **-u <FILE> •** True if **<FILE>** exists and has suid bit set

• **-r <FILE> •** True if **<FILE>** exists and is readable

• **-w <FILE> •** True if **<FILE>** exists and is writable

• **-x <FILE> •** True if **<FILE>** exists and is executable

• **-s <FILE> •** True if **<FILE>** exists and has size bigger than 0

• **-t <fd> •** True if file descriptor **<fd>** is open and refers to a terminal

• **<FILE1> -nt <FILE2> •** True, if **<FILE1>** is newer than **<FILE2>**

• **<FILE1> -ot <FILE2> •** True if **<FILE1>** is older than **<FILE2>**

**String Tests**

• **-z <STRING> •** True if **<STRING>** is empty

• **-n <STRING> •** True if **<STRING>** is not empty; this is the default operation

• **<STRING1> = <STRING2> •** True if the strings are equal

• **<STRING1> != <STRING2> •** True if the strings are not equal

• **<STRING1> < <STRING2> •** True if **<STRING1>** sorts before **<STRING2>** lexicographically;

remember to escape **•**

• **<STRING1> > <STRING2> •** True if **<STRING1>** sorts after **<STRING2>** lexicographically;

remember to escape **•**

**Arithmetic Tests**

• **<INTEGER1> -eq <INTEGER2> •** True if the integers are equal

• **<INTEGER1> -ne <INTEGER2> •** True if the integers are NOT equal

• **<INTEGER1> -le <INTEGER2> •** True if the first integer is less than or equal second one

• **<INTEGER1> -ge <INTEGER2> •** True if the first integer is greater than or equal second one

• **<INTEGER1> -lt <INTEGER2> •** True if the first integer is less than second one

• **<INTEGER1> -gt <INTEGER2> •** True if the first integer is greater than second one

**Misc Syntax**

• **<TEST1> -a <TEST2> •** True if **<TEST1>** and **<TEST2>** are true; -a may also be used as a file

test

• **<TEST1> -o <TEST2> •** True if either **<TEST1>** or **<TEST2>** is true

• **! <TEST> •** True if **<TEST>** is false

• **( <TEST> ) •** Group a test (for precedence); in normal shell-usage, parentheses must be escaped;

use "\(" and "\)"

• **-o <OPTION\_NAME> •** True if the shell option **<OPTION\_NAME>** is set

• **-v <VARIABLENAME> •** True if the variable **<VARIABLENAME>** has been set; use var[n] for

array elements