

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

CS 35L

Software Construction Laboratory

Lecture 2.2

17th January, 2019

Logistics

- ▶ If you are looking for PTE's or wanting to switch labs, continue to write your name on the sheet of paper
- ▶ Assignment 2
 - ▶ Due on Jan 23rd
- ▶ Hardware requirement for Week 8
 - ▶ Seeed Studio BeagleBone Green Wireless Development Board
 - ▶ Office Hours:
 - ▶ 18th Jan - 2:30 to 4:30pm - Boelter Hall 3256S-E

Review - Previous Lab

- ▶ Locale
- ▶ sort, comm and tr
- ▶ Shell Scripting
- ▶ Regular Expressions

Regex Review

- ▶ Quantification
 - ▶ How many times of previous expression?
 - ▶ Most common quantifiers: ?(0 or 1), *(0 or more), +(1 or more)
- ▶ Grouping
 - ▶ Which subset of previous expression?
 - ▶ Grouping operator: ()
- ▶ Alternation
 - ▶ Operators: [] and |
 - ▶ Hello|World [A B C]
- ▶ Anchors
 - ▶ Where?
 - ▶ Characters: ^(beginning) and \$(end)

Regex Exercises

- ▶ Which of the following strings would match the regular expression: `aab?b`
 - ▶ A. `aabb`
 - ▶ B. `aa\nbbb`
 - ▶ C. `aab`

Regex Exercises

- ▶ Which regular expression would match the words “favorite” and “favourite”?

Regex Exercises

- ▶ Which regular expression would match the words "Ggle", "Gogle" and "Google"?
- ▶ Which one would match "Gogle", "Google" and "Goooogle" but not "Ggle"?

Regex Exercises

- ▶ Which regular expression would match any version of the word “Google” that has an even number of o’s?
- ▶ Which regular expression would match any version of the word “Google” that has fewer than 7 O’s?

Regex Exercises

- ▶ Which line(s) would this regular expression match? `“^T.+e$”`
 - ▶ A. The
 - ▶ B. Te
 - ▶ C. Three
 - ▶ D. Then
 - ▶ E. The Two

Regex Exercises

- ▶ Which regular expression(s) would match the words “Ted”, “Ned” and “Sed”?
 - ▶ A. (T|N|S)ed
 - ▶ B. [TNS]ed
 - ▶ C. .ed
 - ▶ D. [L-U]?ed
 - ▶ E. .*ed

Regex Exercises

- ▶ Which regular expression would match all subdirectories within a directory?
 - ▶ A. (T|N|S)ed
 - ▶ B. [T N S]ed
 - ▶ C. .ed
 - ▶ D. [L-U]?ed
 - ▶ E. .*ed

Sed

- ▶ stream editor, modifies the input as specified by the command(s)
- ▶ Can be used for:
 - ▶ Printing specific lines
 - ▶ `sed -n '1p' sedFile.txt`
 - ▶ `sed -n '1,5p' sedFile.txt`
 - ▶ `sed -n '1~2p' sedFile.txt`
 - ▶ Deleting text
 - ▶ `sed '1~2d' sedFile.txt`
 - ▶ Substituting text - `s/regex/replacement/flags`
 - ▶ `sed 's/cat/dog/' file.txt`
 - ▶ `sed 's/cat/dog/g' file.txt`
 - ▶ `sed 's/<[^>]*>//g' a.html`
 - ▶ `sed 's/regExpr/replText/' filename`

Sed – contd..

- ▶ `sed -n 12,18p file.txt`
- ▶ `sed 12,18d file.txt`
- ▶ `sed '1~3d' file.txt`
- ▶ `sed '1,5 s/line/Line/g' file.txt`
- ▶ `sed '/pattern/d' file.txt`

Grep command

- ▶ A Unix command to search files/text for the occurrence of a string of characters that matches a specified pattern
- ▶ Usage:
 - ▶ `grep [option(s)] pattern [file(s)]`
- ▶ `grep -c 'line' file.txt`
- ▶ `grep -n 'line' file.txt`
- ▶ `ls -l | grep ".txt"`

awk

- ▶ awk is more than a command; it's a programming language by itself
- ▶ Utility/language for data extraction
- ▶ awk views a text file as records and fields
- ▶ Usage:
 - ▶ `awk '/search pattern/ {Actions}' file`
- ▶ Examples:
 - ▶ `awk '{print;}' file.txt // print the file line by line; default behaviour`
 - ▶ `awk '/Hello/ {print;}' file.txt // prints lines which matches Hello`
 - ▶ `awk '{print $1,$2;}' file.txt // prints only specific fields`

Assignment 2 - Laboratory

- ▶ Submit 3 files:
 - ▶ Script “buildwords”
 - ▶ Simple text file “lab2.log”
 - ▶ 80 character limit per row
- ▶ Check everything on SEASnet!
 - ▶ Assignments graded on SEASnet servers (eg. Inxsrv07)

Assignment 2 - Laboratory

- ▶ Build a spelling checker for the Hawaiian language
 - ▶ Get familiar with sort, comm and tr commands!
- ▶ Steps:
 - ▶ Download a copy of web page containing basic English-to-Hawaiian dictionary
 - ▶ Extract only the Hawaiian words from the web page to build a simple Hawaiian dictionary. Save it to a file called hwords (site scraping)
 - ▶ Automate site scraping: buildwords script (cat hwnwdseng.htm | buildwords > hwords)
 - ▶ Modify the command in the lab assignment to act as a spelling checker for Hawaiian
 - ▶ Use your spelling checker to check hwords and the lab web page for spelling mistakes

Useful Text Processing Tools

- ▶ `wc`: outputs a one-line report of lines, words, and bytes
- ▶ `head`: extract top of files
- ▶ `tail`: extracts bottom of files
- ▶ `tr`: translate or delete characters
- ▶ `grep`: print lines matching a pattern
- ▶ `sort`: sort lines of text files
- ▶ `sed`: filtering and transforming text

Lab2.log

- ▶ .log is the same as .txt – no difference
- ▶ Ex:
 - ▶ 1. I used wget to download the webpage
 - ▶ 2. I
 - ▶ 3. Answer to #3 here
- ▶ Should read basically like a lab journal
- ▶ Keep things concise!

Lab Hints

- ▶ Run your script on seasnet servers before submitting to CCLE
- ▶ `sed '/patternstart/,/patternstop/d'`
 - ▶ delete patternstart to patternstop, works across multiple lines
will delete all lines starting with patternstart to patternstop
- ▶ The Hawaiian words html page uses `\r` and `\n` for new lines
 - ▶ `od -c hwnwdseng.htm` to see the ASCII characters
- ▶ You can delete blank white spaces such as tab or space using
 - ▶ `tr -d '[:blank:]'`
 - ▶ Use `tr -s` to squeeze multiple new lines into one
- ▶ `sed 's/<[^>]*>//g' a.html` to remove all HTML tags

POSIX Bracket Expressions

Class	Matching characters	Class	Matching characters
<code>[:alnum:]</code>	Alphanumeric characters	<code>[:lower:]</code>	Lowercase characters
<code>[:alpha:]</code>	Alphabetic characters	<code>[:print:]</code>	Printable characters
<code>[:blank:]</code>	Space and tab characters	<code>[:punct:]</code>	Punctuation characters
<code>[:cntrl:]</code>	Control characters	<code>[:space:]</code>	Whitespace characters
<code>[:digit:]</code>	Numeric characters	<code>[:upper:]</code>	Uppercase characters
<code>[:graph:]</code>	Nonspace characters	<code>[:xdigit:]</code>	Hexadecimal digits

Buildwords

- ▶ Hawaiian.html -> buildwords -> hwords
- ▶ Buildwords
 - ▶ Read from STDIN and perform work on input
 - ▶ Output to STDOUT
- ▶ Ex: `$./buildwords < hawaiian.html > hwords`

Assignment 2 - Homework

- ▶ Basic Regular Expressions (BRE) vs Extended Regular Expressions (ERE)
- ▶ Grep -E to work with ERE

Assignment 2 - Homework

Useful grep Options

- ▶ • -l and -L
 - ▶ Suppress normal output; instead print the name of each input file from which output would normally have been printed. The scanning will stop on the first match.
 - ▶ Uppercase L prints all complementary filenames
- ▶ -v
 - ▶ Invert sense of matching

Questions?