cha.2+cha.3

search wildcards

\w for letters and digits

() capture text

+ match one or more than one entities

\t seatch for tabs, separate columns of data

\s search all the white space(inculding space, tabs, and end-of-line)

\r,\n end-of line, in jdeit and pyhon use\n

[] search a range of numbers or letters or mix individual characters;[0-9],[a-z],[A-Z\,]

^ match the beginning of a line

$ match the end of the ine

\* match zero o more times

? produce a minimum match

{} control the number of matches

. any letter, digits, or symbol, excepte the end of line

====cha.4+cha.5===

ls print out a list of files in the working or a specified directionary

ls -a list all th e file, include the hidden files

ls -l print a more detailed list of directory, includes the permission; the number of the item;the owner;the group;the file size;the modifcation date;the file or the directory name;

pwd --print working directionary, to see the absolute path of the working directionary

cd --change directory,to move to another directory

~ refer to your home directory

mkdir --make directory

rmdir --remove directory, remove empty directory;

cp --copy, copy a file

.. refer to the directory that contains the current working directory

./ refer to the current working directory

mv move files

less display the content of text files

man explain what the programe dom which argument should be entered

\* means any character, zero or more;\*.txt,D\*

exit to end your terminal session

nano edit text files; without any arguments will create a new blank file

> sent the output of a programe to a file

cat display the content of the files to the screen, without breake;

cat n.txt>n1.txt write to n1 file

cat n2.txt>>n1.txt write to n1file and keep the first round content

grep extract particular lines from a file; > , redirect to a file;

grep -v extract the line which do not match your search expression

grep -l list the names of files which contain s particualr text pattern, instead of listing each line that match.

grep -n show the line number of the match

grep -h hide the filename of the match

| redirect output from one program to another

curl retrive web content

curl "" > or -o "" dowload the content of the web and store it in a file

===cha.6\_scripting with shell+cha.16===

#! at the beginning of the first line of your script, the shell will send the entire contents of the file to the program

chomd --change mode,modify the permission of the file;chomd u+w

alias create a shortcut to make multiple operations occur with a single command

head -n show the first n lines

tail -n show the last n lines

cut extract columns

cut -c extract which characters

cut -f extract which columns

cut -d "," use commas as the delimiters, instead of tabs;

sort sort lines, by default, sorting starts with the first character of the line and the first column of data

sort -n sort by numeric value rather than alphabetically

sort -r sort in reverse order, z to a

sort -k 3 sort lines based on cloumn 3

sort -t "," use commas for delimiters

uniq -c it must to be sorted before use uniq command;the output includes two columns data: first, how often an element is repreated; second, the name of the element.

agrep approximate search

agrep -d "\>" search the in text blocks divided by > instead of by line endings.

agrep -B -y print out the best match without asking

agrep -l only list filename that contain a match

agrep -i case insensitive search

===cha.7===

variables each variable has few attributes: the name, its type, and the value;

basic variable types

integer (e.g.0, 1,2,3,)

floating point extremely large, or have decimal fraction(e.g.0.9,5.6)

Boolean have two values, true or false

Strings bounded by a pair of straight quotation(')or('');(e.g."anhr")

list one-dimensional arrays are referred to as a list; [2,5,'hi',h]

dictionary it is another type od container for more than one variable;is a collection of name (keys) and an assiciated value

{'ho':78,'hue':35}

mathematical operators

+ addition

- subtraction

\* multipliation

/ division

comparative

== is equal to

!= not equal to

> greater than

< less than

>= greater or equal

<= less or equal

Flow control

if statement for decisions making

for loop repeat execute until some condition is met

while loop a loop without having a predetemined list to cycle through

objects essential components of contemporary programming

dot natation e.g. mybike.color

===cha.8---cha.11===

len() reture the number of elements in the objects

str(), int() and float() convert the variablr types

count() count how many time a particular substring occurs in a string

# add the comments

% %operator for controlling string formatting

input() getting input from user

.replace() and .upper() sanitizing variables

Searching with regular expression

Results = re.search(query,string)

Import the module import re

Working with lists

Creating lists list(MyString) MyList = [2,4,5,]

range() e.g.range(5)-result [0,1,2,3,4,5]

def define the function in the program defore it is used

InFile = open(FileName, 'r') open the file

for Line in InFile read line by line

InFile.close() close the file

OutFile = open(FileName, 'w') open file stream, overwritingexisting file of it exists

OutFile.close() close the outFile when done writing