

# Notes 9

---

## GREP

- **Definition:**

- Grep is used to **search text in given file**. Grep works or search in a line by line basis.

- **Usage:**

- `grep + option + search criteria + file(s)`

- **Common Options:**

- `-i`: Enables case insensitivity (matches regardless of case).
  - `-n`: Displays line number for every line matched.
  - `-E`: Treats the pattern as an extended regular expression.
  - `-G`: Treats the pattern as a basic regular expression.
  - `-v`: Inverts the search (finds lines that do not match the pattern).
  - `-o`: Only displays the matched string.
  - `-c`: Displays the total number of times a pattern is matched.
  - `-w`: Matches only the whole word (exact pattern).
  - `-r` or `-R`: Searches recursively through directories.

- **Examples:**

- **Search any line that contains the word "dracula" in the given file.**
    - `grep 'dracula' ~/Documents/sample_files/Txt/dracula.txt`
  - **Search any line that contains the word "dracula" regardless of the case.**
    - `grep -i 'dracula' ~/Documents/sample_files/Txt/dracula.txt`
  - **Display how many lines contain the matched string.**
    - `grep -c 'Dracula' ~/Documents/sample_files/Txt/dracula.txt`
  - **Search any line that contains the word "dracula" regardless of case and with number line**
    - `grep -in 'dracula' ~/Documents/sample_files/Txt/dracula.txt`
  - **Search for all the lines that do not contain the word 'war'**
    - `grep -v 'war' ~/Documents/sample_files/Txt/war-and-peace.txt`
  - **Search and display only the matched string (pattern)**
    - `grep -o 'pride' ~/Documents/sample_files/Txt/war-and-peace.txt`
  - **Display a list of users with the /bin/bash login shell**
    - `grep -i "/bin/bash" /etc/passwd`
  - **Display your user's information as stored in the /etc/passwd**
    - `grep -i $USER /etc/passwd`
  - **Search for a given strings inside files in a given directory**
    - `grep -iR 'conf' /etc/`
  - **Search and display the total number of times a given word appears in a file**
    - `grep -wc '/bin/bash' /etc/passwd`
  - **The ^ (caret) symbol matches the empty string at the beginning of a line. Search for all the lines that start with a given word**
    - `grep -ni '^dracula' ~/Documents/sample_files/Txt/dracula.txt`
  - **Search for all the lines that ends with the string "nologin"**
    - `grep -n 'nologin$' /etc/passwd`
  - **Search for all the lines that start with a capital letter**

- `grep -n '^[A-Z]' ~/Documents/sample_files/Txt/dracula.txt`
  - Search for **more than one word per line**
    - `grep -Ewn 'horror|love|scare'`  
`~/Documents/sample_files/Txt/dracula.txt`
  - Match only lines containing **IPv4 addresses**
    - `grep -E '[[[:digit:]]{1,3}\.[[[:digit:]]{1,3}\.[[[:digit:]]{1,3}\. [[[:digit:]]{1,3}' ~/Documents/sample_files/Txt/practice.txt`
  - Search all lines that contain a **character repeated 3 times**
    - `grep -E "A{3}" file.txt`
  - Search all lines that contain a **phone number** of the **format 973-111-2222**
    - `grep "[[:digit:]]\{3\}[-][[:digit:]]\{3\}[-][[:digit:]]\{4\}"`  
`~/Documents/sample_files/Csv/contacts.csv`
  - Display only the options of any **command from its man page**
    - `man ls | grep "^[[:space:]]*[[[:punct:]]]"`
  - Search for all the **lines** that contain a **single word "linux"** in all the **files in the system**
    - `grep -niR '^linux$' /`
  - The **period (.) symbol** is a meta-character that **matches any single character**. It searches for all lines that contain a word starting with the letter "d" followed by any 4 characters
    - `grep -n '^d....'` `~/Documents/sample_files/Txt/dracula.txt`
  - **Bracket expressions** allows match a **group of characters** by enclosing them in bracket `[]`. It matches all lines that contain the words "list", "last", or "lost"
    - `grep -n 'l[aio]st'` `~/Documents/sample_files/Txt/dracula.txt`

```

1:aayushmas@Linux:~ $ grep -w '/bin/bash' /etc/passwd
2
aayushmas@Linux:~ $ grep -ni '^dracula' ~/Documents/sample_files/Txt/dracula.txt
106:DRACULA
1038:Dracula, jumping to his feet, said:-
1165:Dracula, as it may help me to understand. To-night he may talk of
1316:Dracula was speaking, there was that in his eyes and in his bearing
1997:Dracula." I suspected him, and determined to test his sincerity.
13810:Dracula we should go by Galatz, or at any rate through Bucharest, so I
aayushmas@Linux:~ $ grep -n 'nologin' /etc/passwd
2:daemon:x:1:1:daemon:/usr/sbin/nologin
3:bin:x:2:bin:/bin:/usr/sbin/nologin
4:sys:x:3:sys:/dev:/usr/sbin/nologin
6:games:x:5:0:games:/usr/games:/usr/sbin/nologin
7:man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8:lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9:mail:x:8:mail:/var/mail:/usr/sbin/nologin
10:news:x:9:news:/var/spool/news:/usr/sbin/nologin
11:uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12:proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13:www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14:backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15:list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16:irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
3:aayushmas@Linux:~ $ grep -n '^[[A-Z]]' ~/Documents/sample_files/Txt/dracula.txt
3:This eBook is for the use of anyone anywhere in the United States and
11:Title: Dracula
13:Author: Bram Stoker
15:Release Date: October, 1995 [eBook #345]
18:Language: English
20:Produced by: Chuck Greif and the Online Distributed Proofreading Team
62:Contents
64:CHAPTER I. Jonathan Harker's Journal
65:CHAPTER II. Jonathan Harker's Journal
66:CHAPTER III. Jonathan Harker's Journal
67:CHAPTER IV. Jonathan Harker's Journal
68:CHAPTER V. Letters-Lucy and Mina
69:CHAPTER VI. Mina Murray's Journal
70:CHAPTER VII. Cutting from "The Dailygraph," 8 August
71:CHAPTER VIII. Mina Murray's Journal
72:CHAPTER IX. Mina Murray's Journal
73:CHAPTER X. Mina Murray's Journal
74:CHAPTER XI. Lucy Westenra's Diary
75:CHAPTER XII. Dr. Seward's Diary
76:CHAPTER XXX. Dr. Seward's Diary
4:aayushmas@Linux:~ $ grep -E '[[[:digit:]]{1,3}].[[:digit:]]{1,3}.[[:digit:]]{1,3}.[[:digit:]]{1,3}' ~/Documents/sample_files/Txt/practice.txt
192.168.0.47
192.168.1.2
aayushmas@Linux:~ $ grep "[[:digit:]]{3}[-][[:digit:]]{3}[-][[:digit:]]{4}" ~/Documents/sample_files/Csv/contacts.csv
James,Butt,"Benton, John B Jr",6649 N Blue Gum St,New Orleans,Orleans,LA,70116,504-621-8927, 504-845-1427,jbutta@gmail.com
Josephine,Darakjy,"Chanay, Jeffrey A Esq",4 B Blue Ridge Blvd,Brighton,Livingston,MI,48116,810-292-9388, 810-374-9840,josephine_darakjy@darakjy.org
Art,Venere,"Chemet, James L Cpa",8 W Cerritos Ave #54,Bridgeport,Gloucester,NJ,8014,856-636-8749, 856-264-4130,art@venere.org
Lenna,Paprocki,Feltz Printing Service,639 Main St,Anchorage,Anchorage,AK,99501,907-385-4412, 907-921-2010,lpaprock@gmail.com
Donette,Foller,Printing Dimensions,34 Center St,Hamilton,Butler,OH,45011,513-570-1893, 513-549-4561,donette.foller@cox.net
Simona,Morasca,"Chapman, Ross E Esq",3 Mcauley Dr,Ashland,Ashland,OH,44805,419-503-2484, 419-800-6759,simona@morasca.com
Mitsue,Tollner,Morlong Associates,7 Eads St,Chicago,Cook,IL,60632,773-573-6914, 773-24-8565,mitsue_tollner@yahoo.com
Leota.Dilliard,Commercial Press,7 W Jackson Blvd,San Jose,Santa Clara,CA,95111,408-7

```

```

1:aayushmas@Linux:~ $ man ls | grep "^[[:space:]]*[[[:punct:]]]"
aayushmas@Linux:~ $ grep -n '[[[:space:]]*[[[:punct:]]]"'
  -sort is specified.
  -a, --all
    .
  -A, --almost-all
  -author
  -b, --escape
  -block-size=SIZE
    '--block-size=M'; see SIZE format
  -B, --ignore-backups
  -c      with -lt: sort by, and show, ctime
          (time of last change of file status
  -C      list entries by columns
  -color[=WHEN]
  -d, --directory
  -D, --dired
  -f      same as -a -U
  -F, --classify[=WHEN]
  -file-type
  -format=WORD
  -full-time
  -g      like -l, but do not list owner
  -group-directories-first
  -n, --no-aversion
3:aayushmas@Linux:~ $ grep -niR '^linux$'
aayushmas@Linux:~ $ grep -n 'l[aio]st' ~/Documents/sample_files/Txt/dracula.txt
210:a siege of three weeks and lost 13,000 people, the casualties of war
256:state that she seemed to have lost her grip of what German she knew, and
311:"word-bearer"--came and listened, and then looked at me, most of them
329:could not but be touched. I shall never forget the last glimpse which I
338:I soon lost sight and recollection of ghostly fears in the beauty of the
367:pointed crags, till these were themselves lost in the distance, where
438:some little time; and at last we saw before us the Pass opening out on
574:At last there came a time when the driver went further afield than he
782:strange stillness over everything; but as I listened I heard as if from
805:last twenty-four hours. I slept till late in the day, and awoke of my
909:certain night of the year--last night, in fact, when all evil spirits
912:went on, "in the region through which you came last night, there can be
940:"Come," he said at last, "tell me of London and of the house which you
1026:After supper I smoked, as on the last evening, and the Count stayed with
1061:imagination must not run riot with me. If it does I am lost. Let me say
1080:round to look for some sticking plaster. When the Count saw my face, his
1370:_Later_--I endorse the last words written, but this time there is no
1436:them except old furniture, dusty with age and moth-eaten. At last,
1469:diary in shorthand all that has happened since I closed it last. It is
1628:rigorously accustomed to wind it the last thing before going to bed. and

```

- **Definition:**

- Awk is a scripting language used for **processing and displaying text files**. Awk can work with a text file or from standard output.
- There are several implementations of Awk: nawk, mawk, gwak, and busybox.
- Awk performs operations line by line.

- **Usage:**

- `awk + options + {awk command} + file + file to save (optional)`

- **Awk Variables:**

Variable	Description
\$0	Whole line
\$1,\$2...\$NF	First, second... last field
NR	Total Number of Records
NF	N number of Fields
OFS	Output Field Separator (default " ")
FS	Input Field Separator (default " ")
ORS	Output Record Separator (default "\n")
RS	Input Record Separator (default "\n")
FILENAME	Name of the file
ARGC	Number of arguments
ARGV	Array of arguments
FNR	File Number of Records
OFMT	Format for numbers (default "%.6g")
RSTART	Location in the string
RLENGTH	Length of match
SUBSEP	Multi-dimensional array separator (default "\034")
ARGIND	Argument Index
ENVIRON	Environment variables
IGNORECASE	Ignore case
CONVFMT	Conversion format
ERRNO	System errors
FIELDWIDTHS	Fixed width fields

- **Examples:**

- Print the **first column** of every line of a file.

- `awk '{print $1}' ~/Documents/sample_files/Csv/cars.csv`

- Print **first field** of /etc/passwd file
  - `awk -F: '{print $1}' /etc/passwd`
- Print the **last field** of the /etc/passwd file
  - `awk -F: '{print $NF}' /etc/passwd`
- Print the **first and last field** of the /etc/passwd
  - `awk -F: '{print $1," = ",$NF}' /etc/passwd`
- Print the **first and 3 field** with line numbers
  - `awk -F: '{print NR,$1,$3}' /etc/passwd`
- Print the **first and 4th field** with a **different field separator**
  - `awk -F: '{OFS="="}{print $1,$4}' /etc/passwd`
- **Start printing a file from a given line** (exclude the first 2 lines)
  - `awk 'NR > 3 { print }' /etc/passwd`
- **Convert the first field to upper/lower case**
  - `awk -F: '{print toupper($1)}' /etc/passwd`
- Prints the **length of a line(record)**
  - `awk '{print length($0)}' /etc/passwd`
- Print **specific fields based on a command output**. For example, the size and file name
  - `ls -lhF Documents/ | awk 'BEGIN { printf "%s\t%s\n", "Size", "Name"} {print $5, "\t", $9}'`
    - BEGIN block is executed once at the start
- Print **specific fields with a head** of the /etc/passwd file
  - `awk -F: 'BEGIN { printf "%s\t\t%s\n", "User", "Shell" } {print $1, "\t", $7}' /etc/passwd`



The screenshot shows a terminal window with five tabs, each displaying an example of the awk command:

- Tab 1:** Shows the use of awk to print every line after the 3rd line from /etc/passwd.
- Tab 2:** Shows the use of awk to print the size and name of files in the Documents directory.
- Tab 3:** Shows the use of awk to print uppercase versions of all user names from /etc/passwd.
- Tab 4:** Shows the use of awk to print the length of each line in /etc/passwd.
- Tab 5:** Shows the use of awk to print the user shell for each user in /etc/passwd.

## SED

- **Definition:**

- Sed is a **stream editor** that perform operations on **files and standard output**. For instance it can **search, find and replace, insert and deletion**.
- By using Sed we can edit files without opening them.

- **Usage:**

- `sed options + sed script + file`

- **Examples:**

- **Replacing a string** in given file **globally**(replace false for true)
  - `sed 's/false/true/g' ~/Documents/sample_files/Json/joke.json`
- Replacing only the **fourth occurrence** per line in a file
  - `sed 's/false/true/4' ~/Documents/sample_files/Json/joke.json`
- Replacing from the **given number occurrence to the rest occurrences** in a file. Start at the second time the word appears and continue to till the end of the file
  - `sed 's/false/true/3g' ~/Documents/sample_files/Json/joke.json`
- Replacing string on a **specific line number**
  - `sed '8 s/false/true/' ~/Documents/sample_files/Json/joke.json`
- Replacing string on a **range of lines**
  - `sed '9,10 s/false/true/' ~/Documents/sample_files/Json/joke.json`
- **To delete a particular line** (line 3)
  - `sed '3d' ~/Documents/sample_files/Json/joke.json`

- `sed '3d' ~/Documents/sample_files/Code/helloworld.py`
- To delete the **last line**
  - `sed '$d' ~/Documents/sample_files/Code/helloworld.py`
- To delete line from **range x to y**
  - `sed '2,4d' ~/Documents/sample_files/Code/helloworld.py`
- To delete from a **given number to last line**
  - `sed '3,$d' ~/Documents/sample_files/Code/helloworld.py`
- To delete **pattern matching line** in a file
  - `sed '/fav/d' ~/Documents/sample_files/Code/helloworld.py`
- To **insert one blank line** after each line
  - `sed G ~/Documents/sample_files/Code/helloworld.c`
- To insert **two blank lines**
  - `sed 'G;G' ~/Documents/sample_files/Code/helloworld.c`
- To **delete blank lines and insert** one blank line after each line
  - `sed '/^$/d;G' ~/Documents/sample_files/Code/helloworld.c`
- **Insert 5 spaces** to the left of every lines
  - `sed 's/^/ /' ~/Documents/sample_files/Code/helloworld.c`

```

1: aayushmas@Linux:~$ sed 's/false/true/g' ~/Documents/sample_files/Json/joke.json
{
  "error": true,
  "category": "Programming",
  "type": "single",
  "joke": "Eight bytes walk into a bar.\nThe bartender asks, \"Can I get you anything?\"\n\"Yeah,\" reply the bytes.\n\"Make us a double.\\"", 
  "flags": {
    "nsfw": true,
    "religious": true,
    "political": true,
    "racist": true,
    "sexist": true,
    "explicit": true
  },
  "id": 34,
  "safe": true,
  "lang": "en"
aayushmas@Linux:~$ 

3: aayushmas@Linux:~$ sed '9,10 s/false/true/' ~/Documents/sample_files/Json/joke.json
{
  "error": false,
  "category": "Programming",
  "type": "single",
  "joke": "Eight bytes walk into a bar.\nThe bartender asks, \"Can I get you anything?\"\n\"Yeah,\" reply the bytes.\n\"Make us a double.\\"", 
  "flags": {
    "nsfw": false,
    "religious": false,
    "political": true,
    "racist": true,
    "sexist": false,
    "explicit": false
  },
  "id": 34,
  "safe": true,
  "lang": "en"
aayushmas@Linux:~$ 

2: aayushmas@Linux:~$ sed '8 s/false/true/' ~/Documents/sample_files/Json/joke.json
{
  "error": false,
  "category": "Programming",
  "type": "single",
  "joke": "Eight bytes walk into a bar.\nThe bartender asks, \"Can I get you anything?\"\n\"Yeah,\" reply the bytes.\n\"Make us a double.\\"", 
  "flags": {
    "nsfw": false,
    "religious": true,
    "political": false,
    "racist": false,
    "sexist": false,
    "explicit": false
  },
  "id": 34,
  "safe": true,
  "lang": "en"
aayushmas@Linux:~$ 

4: aayushmas@Linux:~$ sed '3d' ~/Documents/sample_files/Code/helloworld.py
#!/usr/bin/python3

name=input("Enter your name: ")
color=input("Enter fav color: ")
print(f"Your name is {name} and your fav color is {color}")

hello()
aayushmas@Linux:~$ cat ~/Documents/sample_files/Code/helloworld.py
#!/usr/bin/python3

def hello():
    name=input("Enter your name: ")
    color=input("Enter fav color: ")
    print(f"Your name is {name} and your fav color is {color}")

hello()
aayushmas@Linux:~$ 

```

The screenshot shows a terminal window with four panes, illustrating the conversion of Python code to C code.

- Panels 1 & 2:** Show the original Python code for a "hello" function and its execution.
- Panels 3 & 4:** Show the resulting C code generated by the command `sed '2,4d'` and the final C code after compilation with `gcc`.

```

1: aayushmas@Linux:~ $ sed '$d' ~/Documents/sample_files/Code/helloWorld.py
aayushmas@Linux:~ $ def hello():
    name=input("Enter your name: ")
    color=input("Enter fav color: ")
    print(f"Your name is {name} and your fav color is {color}")

2: aayushmas@Linux:~ $ sed '2,4d' ~/Documents/sample_files/Code/helloWorld.py
aayushmas@Linux:~ $ #!/usr/bin/python3
aayushmas@Linux:~ $ color=input("Enter fav color: ")
aayushmas@Linux:~ $ print(f"Your name is {name} and your fav color is {color}")

3: aayushmas@Linux:~ $ def hello():
    name=input("Enter your name: ")

4: aayushmas@Linux:~ $ sed '3,$d' ~/Documents/sample_files/Code/helloWorld.py
aayushmas@Linux:~ $ #!/usr/bin/python3
aayushmas@Linux:~ $ #include <stdio.h>
aayushmas@Linux:~ $ #include <stdlib.h>
aayushmas@Linux:~ $ #include <string.h>
aayushmas@Linux:~ $ int main()
aayushmas@Linux:~ $ {
aayushmas@Linux:~ $     char name[32];
aayushmas@Linux:~ $     char color[10];
aayushmas@Linux:~ $     printf("Enter name: ");
aayushmas@Linux:~ $     fgets(name, sizeof(name), stdin);
aayushmas@Linux:~ $     name[strcspn(name, "\n")] = 0;
aayushmas@Linux:~ $     printf("Enter color: ");
aayushmas@Linux:~ $     fgets(color, sizeof(color), stdin);
aayushmas@Linux:~ $     color[strcspn(color, "\n")] = 0;
aayushmas@Linux:~ $     printf("Your name is %s and fav color is %s\n", name, color);
aayushmas@Linux:~ $     return 0;
aayushmas@Linux:~ $ }

```

## 1. Explain how to use the pipe (|) for redirecting the output of a command to another.

- **Definition:**
  - The pipe allows us to **redirect** the standard output of a command to the standard input of another.
- **Usage:**
  - `command_1 | command_2 | command_3 | ----- | command_N`
- **Examples:**
  - Use **grep** to look for a string in a **particular man page**
    - `man ls | grep "human-readable"`
  - **Display** only the options of the of **any command from its man page**
    - `man ls | grep "^\[[[:space:]]*[[[:punct:]]"`
  - Display only the **IP addresses** from the output of the ip command
    - `ip addr | grep -Eo '[[[:digit:]]{1,3}\.[[[:digit:]]{1,3}\.[[[:digit:]]{1,3}\.[[[:digit:]]{1,3}'`
  - Display only the **2nd line in a file**
    - `head -2 ~/Documents/sample_files/Lst/users.lst | tail -1`
  - **Parse** a file with **grep and replace** a string in the output
    - `grep -i "honda" ~/Documents/sample_files/Csv/cars.csv | sed 's/Honda/tesla/g'`

```

1:aayushmas@Linux:~ $ man ls | grep "human-readable"
-h, --human-readable
aayushmas@Linux:~ $ ip addr | grep -Eo '[[[:digit:]]{1,3}\.[[[:digit:]]{1,3}\.[[[:digit:]]{1,3}].[[:digit:]]{1,3}'

127.0.0.1
10.0.2.15
10.0.2.255
aayushmas@Linux:~ $ 

2:aayushmas@Linux:~ $ man ls | grep "^[[[:space:]]*[[[:punct:]]]"
-a, --all
-A, --almost-all
--author
-b, --escape
--block-size=SIZE
    '--block-size=M'; see SIZE format below
-B, --ignore-backups
-c      with -lt: sort by, and show, ctime (time of last change of file
-C      list entries by columns
--color[=WHEN]
-d, --directory
-D, --diried
-f      same as -a -U
-F, --classify[=WHEN]
--file-type
--format=WORD
    (-1), verbose (-l), vertical (-C)
--full-time
-g      like -l, but do not list owner
--group-directories-first
-G, --no-group
-h, --human-readable
--si      likewise, but use powers of 1000 not 1024
-H, --dereference-command-line

3:aayushmas@Linux:~ $ head -2 ~/Documents/sample_files/Lst/users.lst | tail -1
LunaByte
aayushmas@Linux:~ $ cat ~/Documents/sample_files/Lst/users.lst
shadow_walker92
LunaByte
crystal_forge
xStormRiderXx
betaWolf
evergreen_7
neonPixel
rustedAnchor
orbit_hopper
CobaltKnight
echo_sentry
midnightMarauder
ZephyrWing
quantumOmega

4:aayushmas@Linux:~ $ grep -i "honda" ~/Documents/sample_files/Csv/cars.csv | sed 's
/Honda/test/g'
tesla Civic;24.0;4;120.0;97.00;2489.;15.0;74;Japan
tesla Civic CVCC;33.0;4;91.00;53.00;1795.;17.5;75;Japan
tesla Civic;33.0;4;91.00;53.00;1795.;17.4;76;Japan
tesla Accord CVCC;31.5;4;98.00;68.00;2045.;18.5;77;Japan
tesla Civic CVCC;36.1;4;91.00;60.00;1800.;16.4;78;Japan
tesla Accord LX;29.5;4;98.00;68.00;2135.;16.6;78;Japan
tesla Civic 1500 gl;44.6;4;91.00;67.00;1850.;13.8;80;Japan
tesla Accord;32.4;4;107.0;72.00;2290.;17.0;80;Japan
tesla Civic 1300;35.1;4;81.00;60.00;1760.;16.1;81;Japan
tesla Prelude;33.7;4;107.0;75.00;2210.;14.4;81;Japan
tesla Accord;36.0;4;107.0;75.00;2205.;14.5;82;Japan
tesla Civic;38.0;4;91.00;67.00;1965.;15.0;82;Japan
tesla Civic (auto);32.0;4;91.00;67.00;1965.;15.7;82;Japan
aayushmas@Linux:~ $ grep -i "honda" ~/Documents/sample_files/Csv/cars.csv
Honda Civic;24.0;4;120.0;97.00;2489.;15.0;74;Japan
Honda Civic CVCC;33.0;4;91.00;53.00;1795.;17.5;75;Japan
Honda Civic;33.0;4;91.00;53.00;1795.;17.4;76;Japan
Honda Accord CVCC;31.5;4;98.00;68.00;2045.;18.5;77;Japan
Honda Civic CVCC;36.1;4;91.00;60.00;1800.;16.4;78;Japan
Honda Accord LX;29.5;4;98.00;68.00;2135.;16.6;78;Japan
Honda Civic 1500 gl;44.6;4;91.00;67.00;1850.;13.8;80;Japan
Honda Accord;32.4;4;107.0;72.00;2290.;17.0;80;Japan
Honda Civic 1300;35.1;4;81.00;60.00;1760.;16.1;81;Japan

```

## 2. Explain how to save the output of a command to a file (>).

- **Definition:**
  - It **redirects or save** the output of a command to a file.
- **Usage:**
  - **command output + > + file**
- **Common Options:**
  - **>**: saves standard output
  - **2>**: saves standard error
  - **&>**: saves both output and error
- **Examples:**
  - **Save the output of a command to a file**
    - **ls -lA ~ > all-files-in-home.txt**
  - **Save the error generated by a command to a file**
    - **ls -lA downloads/ 2> error-of-ls**
  - **Save the error to a file and the success to another**
    - **ls -lA downloads/ Pictures > success.txt 2> error.txt**
  - **Save the error and success to the same file**
    - **ls -lA downloads/ Pictures &> alloutput.txt**
  - **Do not display errors. Send errors to the black hole**
    - **ls -lA downloads/ 2> /dev/null**

The screenshot shows a terminal window with five tabs open, each displaying a different command and its output. The tabs are labeled 1, 2, 3, 4, and 5.

- Tab 1:** Shows the output of `ls -la ~` and `cat all-files-in-home.txt`.
- Tab 2:** Shows the output of `ls -LA downloads/ Pictures > success.txt 2> error.txt` and `cat error.txt`.
- Tab 3:** Shows the output of `ls -LA` and `cat error-of-ls`.
- Tab 4:** Shows the output of `ls -LA` and `cat alloutput.txt`.
- Tab 5:** Shows the output of `ls -LA` and `cat /dev/null`.

### 3. Explain how to append the output of a command to a file.

#### • Definition:

- Append means to **add more to a file instead of overwriting its content**. We use "`>>`" to append.
- When we use `>` on a file that already exist and contains data, we overwrite whatever is already inside the file. For example:
  - `ls -la > allmyfiles.lst`
- In this example, if the file `allmyfiles.lst` had any data prior executing the command, that data will be overwritten by the output of `ls -la`.
- So, if we want to keep the old data? Then we use `>>`. For example:
  - `ls -la >> allmyfiles.lst`
- This will add the output of `ls -la` to the end of the file `allmyfiles.lst`.

#### • Usage:

- `command output + >> + file`

#### • Examples:

- Appends** the current directory list to the end of a file (`append-example.txt`).
  - `ls >> append-example.txt`
- Appends error messages from `list` command to the end of a file (`append-example.txt`).
  - `ls Docu . 2>> append-example.txt`
- Appends lines 1-6 from the last 7 lines of a file to another file.
  - `tail -7 append-example.txt | head -6 >> files-list.txt`

```
1: aayushmas@Linux:~/Documents/sample_files/Code$ cat append-example.txt
#!/bin/bash

hello(){
    read -p "Enter your name: " name
    read -p "Enter fav color: " color
    echo "Your name is $name and your fav color is $color"
}
hello
aayushmas@Linux:~/Documents/sample_files/Code$ ls >> append-example.txt
aayushmas@Linux:~/Documents/sample_files/Code$ cat append-example.txt
#!/bin/bash

hello(){
    read -p "Enter your name: " name
    read -p "Enter fav color: " color
    echo "Your name is $name and your fav color is $color"
}
hello
append-example.txt
helloWorld.c
helloWorld.cpp
helloWorld.py
helloWorld.sh
helloWorld.vala
aayushmas@Linux:~/Documents/sample_files/Code$ █

3: aayushmas@Linux:~/Documents/sample_files/Code$ cd Documents/sample_files/Code/
aayushmas@Linux:~/Documents/sample_files/Code$ ls Docu .
ls: cannot access 'Docu': No such file or directory
.:
append-example.txt helloWorld.cpp helloWorld.sh testing.txt
helloWorld.c helloWorld.py helloWorld.vala
aayushmas@Linux:~/Documents/sample_files/Code$ ls Docu . 2>> append-example.txt
.:
append-example.txt helloWorld.cpp helloWorld.sh testing.txt
helloWorld.c helloWorld.py helloWorld.vala
aayushmas@Linux:~/Documents/sample_files/Code$ cat append-example.txt
#!/bin/bash

hello(){
    read -p "Enter your name: " name
    read -p "Enter fav color: " color
    echo "Your name is $name and your fav color is $color"
}
hello
append-example.txt
helloWorld.c
helloWorld.cpp
helloWorld.py
helloWorld.sh
helloWorld.vala
ls: cannot access 'Docu': No such file or directory
aayushmas@Linux:~/Documents/sample_files/Code$ █
```