Grep Commands Cheat Sheet

Contents

- **Options**
- **Regular Expressions**
- **Basic Text Search**
- **Case Sensitivity**
- **Inverted Match**
- **Matching Whole Words**
- **Display Context**
- **Counting Matches**
- **Line Number and File Information**
- **Multiple Patterns**
- **Regular Expression Search**
- **Output Customization**
- **Fixed String Matching**
- **File Searching**
- **Contextual Examples**

Options

- `-i`: Ignore case (case-insensitive search)
- `-w`: Match whole word
- `-n`: Show line numbers
- `-c`: Count the number of matching lines
- `-v`: Invert match, show lines that do not match
- `-r`: Search files recursively in subdirectories
- `-l`: Show only the filenames of matching files
- `-h`: Do not show filenames in output
- `-e pattern`: Use pattern as the search pattern
- `-f file`: Read the search pattern from a file

- `-E`: Interpret the pattern as an extended regular expression
- `-P`: Interpret the pattern as a Perl-compatible regular expression
- `-m num`: Stop after finding num matches
- `-A num`: Show num lines of trailing context after the match
- `-B num`: Show num lines of leading context before the match
- `-C num`: Show num lines of context before and after the match

Regular Expressions

- `.`: Match any single character
- `^`: Match the beginning of a line
- `$`: Match the end of a line
- `[]`: Match any character inside the brackets
- `[^]`: Match any character NOT inside the brackets
- `()`: Group characters together
- `|`: Match either/or (e.g. `cat|dog`)
- `*`: Match zero or more of the preceding character
- `+`: Match one or more of the preceding character
- `?`: Match zero or one of the preceding character
- `{}`: Match a range of occurrences (e.g. `a{1,3}` matches “a”, “aa”, or “aaa”)

Basic Text Search

Search for a pattern in a file

- Search for a specific word or pattern: `grep 'pattern' file.txt`
- Search across multiple files: `grep 'pattern' file1.txt file2.txt`
- Search recursively in a directory: `grep -r 'pattern' /path/to/directory/`

Case Sensitivity

Perform a case-insensitive search

- Ignore case when matching patterns: `grep -i 'pattern' file.txt`

Inverted Match

Exclude matching lines

- Show lines that do not match the pattern: `grep -v 'pattern' file.txt`

Matching Whole Words

Match entire words only

- Match only whole words in a file: `grep -w 'word' file.txt`

Display Context

Show surrounding lines for matches

- Print 3 lines before each match: `grep -B 3 'pattern' file.txt`
- Print 3 lines after each match: `grep -A 3 'pattern' file.txt`
- Print 3 lines before and after each match: `grep -C 3 'pattern' file.txt`

Counting Matches

Count the number of matching lines

- Display only the count of matching lines: `grep -c 'pattern' file.txt`

Line Number and File Information

Include line numbers or filenames

- Display line numbers for matches: `grep -n 'pattern' file.txt`
- Print filenames with matches: `grep -H 'pattern' file.txt`

Multiple Patterns

Search for multiple patterns

- Match any of several patterns: `grep -e 'pattern1' -e 'pattern2' file.txt`
- Use a file with multiple patterns: `grep -f patterns.txt file.txt`

Regular Expression Search

Enable extended regular expressions

- Use extended regex patterns: `grep -E 'pattern1|pattern2' file.txt`

Use Perl-compatible regular expressions

- Interpret patterns as Perl-compatible regex: `grep -P 'pattern' file.txt`

Output Customization

Highlight matches

- Enable colored output for matches: `grep --color=auto 'pattern' file.txt`

Show only matched parts of lines

- Print only the matching portion of lines: `grep -o 'pattern' file.txt`

Fixed String Matching

Match exact strings

- Search using fixed strings (faster for literals): `grep -F 'exact string' file.txt`

File Searching

List files with matches

- Print only the filenames containing matches: `grep -l 'pattern' *.txt`

List files without matches

- Print filenames that do not contain matches: `grep -L 'pattern' *.txt`

Contextual Examples

Practical Applications

- Search for lines starting with “Error”: `grep '^Error' log.txt`
- Find lines ending with “.com”: `grep '\.com$' emails.txt`
- Match lines containing digits: `grep '[0-9]' file.txt`
- Exclude empty lines: `grep -v '^$' file.txt`