Find Command Cheat Sheet

A concise cheat sheet for the `find` command, covering essential options, conditions, and actions for locating files and directories in Unix-like operating systems. Includes practical examples for common use cases.

Basic Usage and Conditions

Basic Syntax

`find <path> <conditions> <actions>`

Searches for files and directories based on specified criteria, starting from a given path.

Path: The directory to start the search in (e.g., `.`, `/`, `~/Documents`).
Conditions: Criteria to match files (e.g., `-name`, `-type`, `-size`).
Actions: What to do with the matched files (e.g., `-print`, `-exec`, `-delete`).

Name-Based Conditions

`-name <pattern>`	Matches filenames exactly as specified by the pattern. **Example:** `find . -name "*.txt"` (Finds all `.txt` files in the current directory and its subdirectories.)
`-iname <pattern>`	Case-insensitive version of `-name`. **Example:** `find . -iname "*.TXT"` (Finds `.txt`, `.TXT`, `.Txt`, etc.)

User/Group Conditions

`-user <username>`	Finds files owned by the specified username. **Example:** `find /home -user john`
`-group <groupname>`	Finds files belonging to the specified group. **Example:** `find /var/www -group www-data`
`-nouser`	Finds files that are not owned by a valid user (orphaned files). **Example:** `find / -nouser`
`-nogroup`	Finds files that do not belong to a valid group. **Example:** `find / -nogroup`

Type-Based Conditions

`-type f`	Finds regular files. **Example:** `find . -type f`
`-type d`	Finds directories. **Example:** `find . -type d`
`-type l`	Finds symbolic links. **Example:** `find /usr/bin -type l`
`-type b`	Finds block special files. **Example:** `find /dev -type b`
`-type c`	Finds character special files. **Example:** `find /dev -type c`
`-type p`	Finds named pipes (FIFOs). **Example:** `find /tmp -type p`
`-type s`	Finds sockets. **Example:** `find /var/run -type s`

Size and Time Conditions

Size-Based Conditions

`-size <n>[cwbkMG]`	Finds files of the specified size. `n` is a number, and the following suffixes can be used: `c`: bytes `w`: two-byte words `b`: 512-byte blocks (default) `k`: kilobytes `M`: megabytes `G`: gigabytes
`-size +10M`	Finds files larger than 10MB. **Example:** `find . -size +10M`
`-size -10k`	Finds files smaller than 10KB. **Example:** `find /tmp -size -10k`
`-size 1G`	Finds files exactly 1GB in size. **Example:** `find /data -size 1G`

Time-Based Conditions

`-atime <n>`	Finds files last accessed `n` days ago. **Example:** `find . -atime 7` (Finds files accessed 7 days ago.)
`-mtime <n>`	Finds files last modified `n` days ago. **Example:** `find /var/log -mtime +30` (Finds log files modified more than 30 days ago.)
`-ctime <n>`	Finds files whose status was last changed `n` days ago. **Example:** `find . -ctime -1` (Finds files whose status was changed in the last 24 hours.)
`-newer <file>`	Finds files modified more recently than `<file>`. **Example:** `find . -newer reference.txt`
`-anewer <file>`	Finds files which were accessed more recently than `<file>`. **Example:** `find . -anewer reference.txt`
`-cnewer <file>`	Finds files which had their status changed more recently than `<file>`. **Example:** `find . -cnewer reference.txt`

Newer With Time

`-newerat <timestamp>`	Finds files modified more recently than the timestamp. Timestamp should be in a format `YYYY-MM-DD hh:mm:ss`. **Example:** `find . -newerat "2024-01-01 12:00:00"`
`-neweram <timestamp>`	Finds files which were accessed more recently than the timestamp. Timestamp should be in a format `YYYY-MM-DD hh:mm:ss`. **Example:** `find . -neweram "2024-01-01 12:00:00"`
`-newerc <timestamp>`	Finds files which had their status changed more recently than the timestamp. Timestamp should be in a format `YYYY-MM-DD hh:mm:ss`. **Example:** `find . -newerc "2024-01-01 12:00:00"`

Actions and Advanced Options

Action-Based Options

`-print`	Prints the matched file or directory path to standard output (default action). **Example:** `find . -name "*.log" -print`
`-exec <command> {} ;`	Executes the specified command on each matched file. `{}` is replaced by the file path, and `\;` terminates the command. **Example:** `find . -name "*.tmp" -exec rm {} \;` (Deletes all `.tmp` files.)
`-ok <command> {} ;`	Similar to `-exec`, but prompts the user for confirmation before executing the command on each file. **Example:** `find . -name "*.txt" -ok rm {} \;`
`-delete`	Deletes the matched files or directories (use with caution!). **Example:** `find . -type f -name "*.bak" -delete`

Combining Conditions

`\( <condition1> -and <condition2> \)` or `<condition1> -a <condition2>`	Finds files that satisfy both `condition1` and `condition2`. **Example:** `find . \( -type f -and -name "*.txt" \)`
`\( <condition1> -or <condition2> \)` or `<condition1> -o <condition2>`	Finds files that satisfy either `condition1` or `condition2` (or both). **Example:** `find . \( -size +1M -or -name "*.log" \)`
`! <condition>` or `-not <condition>`	Finds files that do not satisfy the specified condition. **Example:** `find /home -not -user john`

Other Useful Options

`-depth <levels>`	Processes the contents of each directory at the specified level. Useful for controlling search depth. **Example:** `find . -depth 1` (Searches only within the current directory, not subdirectories.)
`-maxdepth <levels>`	Descends at most `levels` levels of directories below the starting point. **Example:** `find . -maxdepth 3 -type f` (Searches files up to 3 levels deep.)
`-mindepth <levels>`	Does not apply any tests or actions at levels less than `levels`. **Example:** `find . -mindepth 2 -name "*.txt"` (Searches for .txt files starting from the second level.)
`-regex <pattern>`	Uses a regular expression to match the entire file path. **Example:** `find . -regex ".*/[A-Z].*\.txt"` (Files with a capital letter directory `.txt` extension.)

Practical Examples

Common Use Cases

Finding and deleting empty directories:
`find . -type d -empty -delete`

Finding files modified in the last hour:
`find . -type f -mmin -60`

Finding setuid files:
`find / -perm -4000`

Finding files without execute permissions for others:
`find . -type f ! -perm -o+x`

Finding files that have been accessed in the last week:
`find . -atime -7`

Finding files owned by a specific user and group:
`find /home -user john -group developers`

Advanced Examples

Finding and compressing files older than 30 days:
`find . -type f -mtime +30 -exec gzip {} \;`

Listing all files in the current directory sorted by size:
`find . -type f -printf '%s %p\n' | sort -nr | head`

Finding all files bigger than 10MB and prompting before deleting:
`find . -type f -size +10M -ok rm {} \;`

Executing a script on each found file:
`find . -name "*.py" -exec python3 {} \;`

Handling Errors

Suppressing error messages (e.g., permission denied):
`find . -name "*.txt" 2>/dev/null`

Logging errors to a file:
`find / -name "*.conf" 2>errors.log`