

# Riferimento Rapido find

Ricerca file per nome, tipo, dimensione, tempo, permessi e azioni

## Ricerca Base

### Eseguire find

```
find . # elenca tutti i file
ricorsivamente
find /var/log # ricerca da percorso specifico
find . -name "*.txt" # trova per nome
find / -name "config" 2>/dev/null # sopprimi errori permesso
find dir1 dir2 -name "*.py" # cerca in più directory
```

## Sintassi Generale

<b>find</b> [path...] [tests] [actions]	Forma generale — percorsi, poi test, poi azioni
<b>path</b>	Directory di partenza (default: directory corrente)
<b>test</b>	Condizione per filtrare file (-name, -type, ecc.)
<b>action</b>	Cosa fare con le corrispondenze (-print, -exec, ecc.)
<b>Azione predefinita</b>	<b>-print</b> se nessuna azione è specificata

## Pattern di Nome

### Corrispondenza Nome

```
find . -name "*.log" # glob case-sensitive
find . -iname "readme*" # glob case-insensitive
find . -name "*.py" -o -name "*.js" # OR: file Python o JS
find . -path "*/src/*.ts" # corrispondenza percorso completo
find . -regex '.*\\.\\(py\\|js\\)' # regex POSIX su percorso completo
```

## Test Nome

<b>-name pattern</b>	Basename corrisponde a glob shell (case-sensitive)
<b>-iname pattern</b>	Basename corrisponde a glob (case-insensitive)
<b>-path pattern</b>	Percorso completo corrisponde a glob shell
<b>-ipath pattern</b>	Percorso completo corrisponde a glob (case-insensitive)
<b>-regex pattern</b>	Percorso completo corrisponde a espressione regolare
<b>-iregex pattern</b>	Percorso completo corrisponde a regex (case-insensitive)

## Filtri per Tipo

### Filtra per Tipo

```
find . -type f # solo file regolari
find . -type d # solo directory
find . -type l # link simbolici
find . -type f -name "*.sh" # combina tipo + nome
```

## Tipi di File

<b>-type f</b>	File regolare
<b>-type d</b>	Directory
<b>-type l</b>	Link simbolico
<b>-type b</b>	Dispositivo a blocchi
<b>-type c</b>	Dispositivo a caratteri
<b>-type p</b>	Pipe con nome (FIFO)
<b>-type s</b>	Socket
<b>-empty</b>	File o directory vuota

## Dimensione e Tempo

### Esempi Dimensione e Tempo

```
find . -size +100M # più grande di 100 MB
find . -size -1k # più piccolo di 1 KB
find . -mtime -7 # modificato negli ultimi 7 giorni
find . -mmin -30 # modificato negli ultimi 30 minuti
find . -newer reference.txt # più recente del file di riferimento
```

## Test Dimensione e Tempo

<b>-size +/-Nc</b>	Dimensione in byte (c), kilobyte (k), megabyte (M), gigabyte (G)
<b>-mtime +/-N</b>	Modificato N*24 ore fa (+più vecchio, -più recente)
<b>-atime +/-N</b>	Acceduto N*24 ore fa
<b>-ctime +/-N</b>	Stato cambiato N*24 ore fa
<b>-mmin +/-N</b>	Modificato N minuti fa
<b>-newer file</b>	Modificato più di recente del file
<b>-newermt date</b>	Modificato dopo la stringa data (GNU)

## Permessi

### Esempi Permessi

```
find . -perm 644 # permessi esatti: rw-r--r--
find . -perm -u+x # utente ha bit execute
find . -perm /o+w # altri hanno write (qualsiasi)
find . -user root # di proprietà di root
find . -group www-data -type f # di proprietà del gruppo
```

## Test Permessi

<b>-perm mode</b>	Corrispondenza esatta dei permessi
<b>-perm -mode</b>	Tutti i bit specificati sono impostati
<b>-perm /mode</b>	Almeno un bit specificato è impostato
<b>-user name</b>	Di proprietà dell'utente (nome o UID)
<b>-group name</b>	Di proprietà del gruppo (nome o GID)
<b>-nouser</b>	Nessun utente corrispondente in /etc/passwd
<b>-nogroup</b>	Nessun gruppo corrispondente in /etc/group

## Azioni

### Esempi di Azioni

```
find . -name "*.log" -print # stampa percorsi (default)
find . -name "*.tmp" -delete # elimina file corrispondenti
find . -type f -ls # lista dettagliata
find . -name "*.txt" -print0 # output delimitato da null
find . -type f -printf "%p %s\n" # formato personalizzato (GNU)
```

## Riferimento Azioni

<b>-print</b>	Stampa percorso (delimitato da newline)
<b>-print0</b>	Stampa percorso (delimitato da null, sicuro per xargs)
<b>-ls</b>	Stampa dettagli file (come <b>ls -dils</b> )
<b>-delete</b>	Elimina file corrispondenti (implica <b>-depth</b> )
<b>-printf format</b>	Formato output personalizzato (GNU): <b>%p</b> percorso, <b>%s</b> dimensione
<b>-fprint file</b>	Scrivi percorsi su file
<b>-quit</b>	Esce dopo la prima corrispondenza

## Combinare Test

### Operatori Logici

```
find . -name "*.py" -type f # AND implicito
find . -name "*.py" -a -size +10k # AND esplicito
find . -name "*.py" -o -name "*.js" # OR
find . ! -name "*.log" # NOT
find . \( -name "*.py" -o -name "*.js" \) -type f
```

## Riferimento Operatori

<b>expr1 expr2 / expr1 -a expr2</b>	AND — entrambi devono essere veri (default)
<b>expr1 -o expr2</b>	OR — almeno uno deve essere vero
<b>! expr / -not expr</b>	NOT — nega l'espressione
<b>\( expr \)</b>	Raggruppa espressioni (escape parentesi nella shell)
<b>Ordine valutazione</b>	Da sinistra a destra; <b>-a</b> lega più stretto di <b>-o</b>

## Exec e Delete

### Esempi Exec

```
find . -name "*.sh" -exec chmod +x {} \;
find . -name "*.log" -exec rm {} +
find . -type f -exec grep -l "TODO" {} +
find . -name "*.bak" -ok rm {} \; # chiede conferma
find . -name "*.tmp" -print0 | xargs -0 rm
```

## Riferimento Exec

<b>-exec cmd {} \;</b>	Esegui cmd una volta per file ({} = percorso file)
<b>-exec cmd {} +</b>	Esegui cmd con più file insieme (più veloce)
<b>-ok cmd {} \;</b>	Come <b>-exec</b> ma chiede conferma
<b>-execdir cmd {} \;</b>	Esegui cmd dalla directory del file
<b>xargs -0</b>	Abbinare a <b>-print0</b> per elaborazione batch sicura
<b>-delete</b>	Elimina file; processa prima i più profondi

## Profondità e Pruning

### Esempi Profondità e Prune

```
find . -maxdepth 1 -type f # solo directory corrente
find . -mindepth 2 -name "*.py" # salta livello superiore
find . -name ".git" -prune -o -print # salta dir .git
find . -depth -name "*.tmp" -delete # processa figli prima
```

## Opzioni Profondità

<b>-maxdepth N</b>	Scendi al massimo N livelli (0 = solo percorso iniziale)
<b>-mindepth N</b>	Non applicare test a livelli inferiori a N
<b>-depth</b>	Processa contenuto directory prima della directory stessa
<b>-prune</b>	Non scende nella directory corrispondente
<b>-mount / -xdev</b>	Non attraversa i confini del filesystem
<b>-follow / -L</b>	Segue i link simbolici

## Pattern Comuni

### One-Liner

```
find . -name "*.pyc" -delete # pulisci bytecode Python
find . -type f -size 0 -delete # rimuovi file vuoti
find . -mtime +30 -name "*.log" -delete # elimina log vecchi
find . -type f -name "*.md" | wc -l # conta file Markdown
find . -type d -empty -delete # rimuovi dir vuote
```

## Ricette

<b>Trova file più grandi</b>	<b>find . -type f -printf '%s %p\n'   sort -rn   head</b>
<b>Trova duplicati per nome</b>	<b>find . -type f   awk -F/ '{print \$NF}'   sort   uniq -d</b>
<b>Rinomina estensione</b>	<b>find . -name '*.txt' -exec rename 's/.txt/.md/' {} +</b>
<b>Trova link simbolici rotti</b>	<b>find . -xtype l</b>
<b>Archivia file recenti</b>	<b>find . -mtime -7 -print0   tar czf recent.tar.gz --null -T -</b>
<b>Cerca in file codice</b>	<b>find . -name '*.py' -exec grep -l 'pattern' {} +</b>