

find クイックリファレンス

名前、タイプ、サイズ、日時、パーミッション、アクションによるファイル検索

基本的な検索

find の実行

```
find . # 全ファイルを再帰的に一覧表示
find /var/log # 特定のパスから検索
find . -name "*.txt" # 名前で検索
find / -name "config" 2>/dev/null # パーミッションエラーを抑制
find dir1 dir2 -name "*.py" # 複数ディレクトリを検索
```

構文の概要

find [path...] [tests] [actions]	一般形式 - パス、テスト、アクションの順
path	開始ディレクトリ (デフォルト: カレントディレクトリ)
test	ファイルをフィルタする条件 (-name、-type など)
action	一致したファイルへの操作 (-print、-exec など)
デフォルトアクション	アクションが指定されない場合は -print

名前パターン

名前マッチング

```
find . -name "*.log" # 大文字小文字を区別するグロブ
find . -iname "readme*" # 大文字小文字を区別しないグロブ
find . -name "*.py" -o -name "*.js" # OR: Python または JS ファイル
find . -path "*/src/*.ts" # フルパスで一致
find . -regex '.*\.(py|js)\.' # フルパスへの POSIX 正規表現
```

名前テスト

-name pattern	ベース名がシェルグロブに一致 (大文字小文字を区別)
-iname pattern	ベース名がグロブに一致 (大文字小文字を区別しない)
-path pattern	フルパスがシェルグロブに一致
-ipath pattern	フルパスがグロブに一致 (大文字小文字を区別しない)
-regex pattern	フルパスが正規表現に一致
-iregex pattern	フルパスが正規表現に一致 (大文字小文字を区別しない)

タイプフィルター

タイプでフィルタ

```
find . -type f # 通常ファイルのみ
find . -type d # ディレクトリのみ
find . -type l # シンボリックリンク
find . -type f -name "*.sh" # タイプ + 名前の組み合わせ
```

ファイルタイプ

-type f	通常ファイル
-type d	ディレクトリ
-type l	シンボリックリンク
-type b	ブロックデバイス
-type c	キャラクターデバイス
-type p	名前付きパイプ (FIFO)
-type s	ソケット
-empty	空のファイルまたはディレクトリ

サイズと日時

サイズと日時の例

```
find . -size +100M # 100 MB より大きい
find . -size -1k # 1 KB より小さい
find . -mtime -7 # 過去 7 日以内に變更
find . -mmin -30 # 過去 30 分以内に變更
find . -newer reference.txt # 参照ファイルより新しい
```

サイズと日時テスト

-size +/-Nc	バイト(c)、キロバイト(k)、メガバイト(M)、ギガバイト(G)のサイズ
-mtime +/-N	N*24 時間前に變更 (+は古い、-は新しい)
-atime +/-N	N*24 時間前にアクセス
-ctime +/-N	N*24 時間前にステータス變更
-mmin +/-N	N 分前に變更
-newer file	ファイルより新しく變更された
-newermt date	日付文字列以降に變更 (GNU)

パーミッション

パーミッションの例

```
find . -perm 644 # 正確なパーミッション: rw-r--r--
find . -perm -u+x # ユーザーに実行ビットがある
find . -perm /o+w # 他者に書き込み権限 (いずれか一致)
find . -user root # root が所有
find . -group www-data -type f # グループが所有
```

パーミッションテスト

-perm mode	正確なパーミッション一致
-perm -mode	指定したビットがすべて設定されている
-perm /mode	指定したビットのいずれかが設定されている
-user name	ユーザーが所有 (名前または UID)
-group name	グループが所有 (名前または GID)
-nouser	/etc/passwd に一致するユーザーがない
-nogroup	/etc/group に一致するグループがない

アクション

アクションの例

```
find . -name "*.log" -print # パスを表示 (デフォルト)
find . -name "*.tmp" -delete # 一致したファイルを削除
find . -type f -ls # 詳細一覧
find . -name "*.txt" -print0 # null 区切り出力
find . -type f -printf "%p %s\n" # カスタムフォーマット (GNU)
```

アクションリファレンス

-print	パスを表示 (改行区切り)
-print0	パスを表示 (null 区切り、xargs に安全)
-ls	ファイルの詳細を表示 (ls -dils 相当)
-delete	一致したファイルを削除 (-depth を暗黙的に含む)
-printf format	カスタム出力フォーマット (GNU): %p パス、 %s サイズ、 %t 日時
-fprint file	パスをファイルに書き込む
-quit	最初の一致後に終了

テストの組み合わせ

論理演算子

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

演算子リファレンス

expr1 expr2 / expr1 -a expr2	AND - 両方が true である必要がある (デフォルト)
expr1 -o expr2	OR - いずれかが true である必要がある
! expr / -not expr	NOT - 式を否定
\(expr \)	式をグループ化 (シェルで括弧をエスケープ)
評価順	左から右へ; -a は -o より優先度が高い

exec と削除

exec の例

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

exec リファレンス

-exec cmd {} \;	ファイルごとに cmd を実行 ({} = ファイルパス)
-exec cmd {} +	複数ファイルで cmd をまとめて実行 (高速)
-ok cmd {} \;	-exec と同様だが確認を求める
-execdir cmd {} \;	ファイルのディレクトリから cmd を実行
xargs -0	安全なバッチ処理のために -print0 と組み合わせる
-delete	ファイルを削除; 深い場所から先に処理

深さと枝刈り

深さと枝刈りの例

```
find . -maxdepth 1 -type f # カレントディレクトリのみ
find . -mindepth 2 -name "*.py" # トップレベルをスキップ
find . -name ".git" -prune -o -print # .git ディレクトリをスキップ
find . -depth -name "*.tmp" -delete # 子を先に処理
```

深さオプション

-maxdepth N	最大 N レベルまで降りる (0 = 開始パスのみ)
-mindepth N	N レベル未満ではテストを適用しない
-depth	ディレクトリ自体の前にディレクトリ内容を処理
-prune	一致したディレクトリには降りない
-mount / -xdev	ファイルシステムの境界を越えない
-follow / -L	シンボリックリンクを辿る

よくあるパターン

ワンライナー

```
find . -name "*.pyc" -delete # Python バイトコードを削除
find . -type f -size 0 -delete # 空ファイルを削除
find . -mtime +30 -name "*.log" -delete # 古いログを削除
find . -type f -name "*.md" | wc -l # Markdown ファイルを数える
find . -type d -empty -delete # 空ディレクトリを削除
```

find クイックリファレンス

レシピ

最大ファイルを探す	<pre>find . -type f -printf '%s %p\n' sort -rn head</pre>
名前で重複を探す	<pre>find . -type f awk -F/ '{{print \$NF}}' sort uniq -d</pre>
拡張子を変更	<pre>find . -name '*.txt' -exec rename 's/.txt/.md/' {} +</pre>
壊れたシンボリックリンクを探す	<pre>find . -xtype l</pre>
最近のファイルをアーカイブ	<pre>find . -mtime -7 -print0 tar czf recent.tar.gz --null -T -</pre>
コードファイルを検索	<pre>find . -name '*.py' -exec grep -l 'pattern' {} +</pre>