

\$ find

► Syntax: `$ find . -name *.txt`

Start-Verzeichnis Suchkriterien

Das Programm `find` durchsucht das Start-Verzeichnis (welches auch immer der 1. Parameter ist) nach bestimmten Suchkriterien und liefert die Suchergebnisse zurück.

Suche nach Typ des Eintrags im Dateisystem:

Die `-type` Option gibt an, wonach `find` suchen soll:

Parameter	Bedeutung
<code>d</code>	"directory" Ordner
<code>f</code>	"file" Datei
<code>l</code>	"link" Link / Verknüpfung auf Datei
<code>s</code>	"socket" Pseudo-Datei für Netzwerk-Schnittstellen
<code>c</code>	"character Device" } Geräte-Dateien
<code>b</code>	"block Device" }
<code>p</code>	"pipe" Linux - FIFO

Standardmäßig sucht find nach allen möglichen Einträgen. Erst durch setzen der Option `-type` wird die Suche eingeschränkt.

Beispiel: Suche nach dem Ordner
„Hausaufgaben“ im Home-Verzeichnis

```
$ find ~ -type d -name Hausaufgaben
```

Suche nach Namen

Am häufigsten wird find verwendet, um eine Datei oder einen Ordner mit einem bestimmten Namen zu finden.

`-name` ist die dafür benötigte Suchfunktion.

Bsp.: suche nach der Datei „Hintergrund.png“

```
$ find . -name Hintergrund.png
```

Damit wird die Datei „Hintergrund.png“ gefunden, die Datei „hintergrund.png“ aber nicht, da find case-sensitiv ist und zwischen Groß- & Kleinschreibung unterscheidet.

Um Groß- und Kleinschreibung zu ignorieren, kann der Parameter `-iname` benutzt werden.

\$ find . -iname Hintergrund.png

würde somit die Dateien „Hintergrund.png“
als auch „hintergrund.png“ finden

Falls vom Dateinamen aber nur teile bekannt
sind, kann der Asterisk * als Platzhalter
für ein beliebiges Zeichen verwendet werden.

\$ find ~/Bilder/ -name *.png

sucht nach allen .png - Bildern im
Bilder - Ordner.

Hinweis:

- name *.png und

- name '*.png'

verhalten sich an der stelle idemisch.
Der Dateiname kann in Anführungszeichen
stehen, muss aber nicht.

Löschen von Suchergebnissen

Die Option -delete löscht gefundene
Dateien und Ordner (aber nur wenn diese leer sind)

\$ find ~/Downloads -name *.sh -delete

Das Löschen von nicht-leeren Ordnern.

Es ist sehr wichtig, dass `-delete` hinter dem Suchausdruck angegeben wird. Find versteht den Aufruf wie eine Kommandozeile von links nach rechts aus und würde, wenn `-delete` vor dem Suchausdruck steht, alles im Suchverzeichnis löschen.

Erweiternde Optionen für den Namen

- Dateien finden, die nicht so heißen:

Suche nach allen Dateien, die keine .png Bilder sind:

```
$ find ~/Bilder -not -name *.png
```

Alternativ kann auch ein Ausrufezeichen ! als Nicht-operator verwendet werden:

```
$ find ~/Bilder ! -name *.png
```

- Weitere Boolesche Operatoren:

Der Und-operator: `-a` oder `-and`

Der Oder-operator: `-o` oder `-or`

Bsp.:

```
$ find . -name '*.jpg' -o -name '*.png'
```

=> Suche alle jpg und png Bilder

Wenn mehr als zwei Operatoren verwendet werden, können Klammern benutzt werden. Diese müssen wegen der Shell aber mit einem Backslash \ maskiert werden.

```
$ find ~ / Bilder ! \ ( -name '*.png'  
-o -name '*.jpg' \) -delete
```

=> Lösche alle Bilder, die nicht als .png oder als jpg gespeichert sind.

Suche mit regulären Ausdrücken:

Reguläre Ausdrücke sind ein mächtiges Werkzeug, mit dem Zeichenketten (für Dateinamen) und Ordner) beschrieben werden.

An dieser Stelle wird das Verständnis für den Aufbau und die Verwendung von Regex vorausgesetzt.

Der `-regex` Parameter erlaubt es einen Regex für die Suche anzugeben:

```
$ find . -regex '[0-9]{5,9}\.'
```

=> finde alle Dateinamen die aus 5 bis 9 Ziffern bestehen, gefolgt von einem Punkt und beliebiger Dateierweiterung.

Wenn die Groß- & Kleinschreibung ignoriert werden soll, kann das durch die Verwendung von `-iregex` erzielt werden:

```
$ find . -iregex '[a-z]\.txt'
```

Da es diverse Regex - Unterarten gibt, kann mit `-regex type` eine angegeben werden. Zur Auswahl stehen:

- findutils
- emacs
- gnu - awk
- grep
- egrep
- posix - awk
- posix - basic
- posix - egrep
- posix - extended

Suche nach Dateigröße

Der `-size` Parameter ermöglicht ein Berücksichtigen der Dateigröße beim Suchen.

Dabei gibt es verschiedene Möglichkeiten, wie dieser Parameter eingesetzt wird.

Syntax: `$ find . -size unnn`

↑
hier steht die Dateigröße mit einem Komparator - Prefix.

1.) Dateigröße:

Für die Angabe der Dateigröße gibt es ein paar Suffixe die die Suche erleichtern sollen:

- `b` 512-Byte-Block (default)
- `c` bytes
- `w` 2-byte-Wörter
- `k` Kilobytes
- `M` Megabytes
- `G` Gigabytes

Bsp.: `$ find . -name '*.png' -size 5M`

⇒ Suche nach 5 MB großen .png Dateien

2.) Comparator:

Meistens möchte man nicht nach Dateien mit einer bestimmten Größe suchen sondern Dateien finden die eine Größe über- bzw. unterschreiten.

Dafür kann vor die Größenangabe ein + oder ein - angehängt werden.

- + : Suche nach Dateien die größer als die angegebene Dateigröße sind.
- - : Suche nach Dateien die kleiner als die angegebene Dateigröße sind.

Bsp.: `$ find . -name '*.png' -size +5M`

=> Suche nach .png Dateien die größer als 5 MB sind.

Suche nach Inhaber

Mit `-user` und `-group` kann ein Benutzer bzw. eine Benutzergruppe berücksichtigt werden

`$ find / -user colord`

=> findet /proc Einträge von colord

- `nowuser` und -`nowroup` liefern alle Ergebnisse, die nicht mit dem darauf folgenden Nutzer bzw. der darauf folgenden Gruppe assoziiert werden.

\$ `find /home/user1 -nowuser user1` \Rightarrow alle Dateien im Home-Verzeichnis die einem anderen Benutzer gehören.

Suche nach Berechtigung

- `perm` nimmt eine zu berücksichtigende Datei - Berechtigung entgegen. Die Berechtigung wird im selben Stil wie bei dem Programm `chmod`, also als dreistellige Oktalziffer angegeben.

\$ `find /usr -perm 755`

\Rightarrow findet ausführbare Binaries in /usr, die mit read Rechten installiert wurden.

Leere Dateien & Verzeichnisse

Mid -`empty` kann nach leeren Dateien und Verzeichnissen gesucht werden.

\$ `find ~ -empty -delete`

\Rightarrow finde alle leeren Dateien und Verzeichnisse und lösche diese.

Suche nach Zeitstempel

► - newer file

=> Suchergebnisse, deren Zeitstempel für die letzte Änderung neuer als der von file ist.

► geändert vor n Tagen

- - atime : letzter Zugriff
- - ctime : letzte Änderung der Berechtigungen
- - mtime : letzte Dateiänderung

\$ find . -mtime 5

=> finde alles, was vor 5 Tagen geändert wurde

\$ find . -mtime -5

=> finde alles, was vor weniger als 5 Tagen geändert wurde

\$ find . -mtime +5

=> finde alles, was vor mehr als 5 Tagen geändert wurde

Wenn man von Beginn des Tages an, und nicht in Vielzahlen von 24h Blöcken suchen möchte, kann man den Parameter

-daystart

zusätzlich angeben.

► geändert vor n Minuten

Die Parameter -amin, -amin und -amin können genau so eingesetzt werden wie deren im vorherigen Abschnitt beschriebenen Äquivalente. Die darauf folgende Zahl gibt den Zeitsdempel aber nicht in Tagen, sondern in Minuten an.

Ein Programm ausführen

Mit dem -exec Parameter können die Suchergebnisse einem Programm als Parameter übergeben werden.

Bsp.: Suche innerhalb von Dateien

```
$ find . -name "*.txt" \
  -exec grep "text-to-find" {} \;
```