tree Directory tree listing

START HERE

tree
Show this directory as a tree

`tree -d`
Directories only

`tree <dir>`
Show another directory

`tree -I '<pattern>'`
Hide matching names

`tree -L 2`
Limit depth to 2 levels

`tree -h`
Show sizes, readable

`tree -a`
Include hidden files

`tree --gitignore`
Respect .gitignore

CONCEPTS

tree skips dotfiles unless you pass -a, and prints a count of directories and files at the end.

Walk -L, -a, -x, -I
↓ `filter`
Filter -P, -I, --prune
↓ `sort`
Sort -t, -r, --dirsfirst
↓ `decorate`
Print -h, -p, -D, -J, -H

-P and -I take shell-style globs, not regexes; separate alternatives with | and quote the pattern.

-P only filters files; add --matchdirs to match directory names, and --prune to drop dirs left empty.

LISTING

tree
Print the current directory as a tree

`tree <dir> ...`
Several directories at once

`tree -L <n>`
Descend at most n levels

`tree -a`
Show hidden files too

`tree -d`
List directories only

`tree -f`
Print the full path of each entry

`tree -l`
Follow symlinks to directories

`tree -x`
Stay on one filesystem

`tree --noreport`
Omit the final count line

LISTING (cont.)

`tree --filelimit <n>`
Skip dirs holding more than n entries

`tree --condense`
Merge single-child dir chains

`tree -i`
No indentation lines (flat list)
Pair with -f for a plain list of paths.

`tree -F`
Append type markers (/ * @ |)

`tree -Q`
Quote filenames

FILTERING

`tree -I '<glob>'`
Hide matching files and dirs

`tree -I 'node_modules|dist'`
Hide several names

`tree -P '<glob>'`
Show only matching files

`tree -P '*.py' --prune`
Matching files, no empty dirs

`tree --gitignore`
Apply .gitignore rules

`tree --gitfile <file>`
Use another ignore file

`tree --matchdirs`
Apply -P to directory names

`tree --ignore-case`
Case-insensitive -P and -I

`tree --prune`
Drop empty directories

DETAILS

`tree -h`
Human-readable sizes

`tree -s`
Sizes in bytes

DETAILS (cont.)

`tree --du`
Directory totals (implies -s)

`tree --si`
Sizes in powers of 1000

`tree -p`
Show permissions

`tree -u / -g`
Show owner / group

`tree -D`
Show modification date

`tree --timefmt '<fmt>'`
Date format (strftime), implies -D

`tree -pugh`
Mode, owner, group and size

`tree --inodes`
Show inode numbers

`tree --metafirst`
Metadata before the tree lines

SORTING

`tree --dirsfirst`
Directories before files

`tree --filesfirst`
Files before directories

`tree -t`
Sort by modification time

`tree -r`
Reverse the order

`tree -v`
Natural version sort (v2 before v10)

`tree --sort <key>`
name, size, mtime, ctime, version, none

`tree -U`
Unsorted, directory order

OUTPUT

`tree -J`
JSON output

`tree -X`
XML output

`tree -H <baseHREF>`
HTML page with links

`tree -H . -T '<title>' -o <file>`
HTML with title to a file

`tree -o <file>`
Write output to a file

`tree -n`
Never colourise

`tree -C`
Force colour, even in a pipe
Needed for `tree -C | less -R`.

`tree -A`
ANSI line-drawing characters

`tree --charset <name>`
Output charset (e.g. ASCII)

`tree --hyperlink`
Clickable OSC 8 file links

`tree --fromfile .`
Build a tree from paths on stdin

RECIPES

Project overview

```
tree -a -L 3 --gitignore -I
'.git' --dirsfirst
# -I '.git': --gitignore does
not hide the .git dir itself
```

Biggest directories

```
tree --du -h -L 2 --sort size -r
# totals per directory, largest
first
```

Tree of a file list

```
git ls-files | tree --fromfile .
find . -name '*.md' | tree --
fromfile .
```

Share it as JSON or HTML

```
tree -J -L 2 src > tree.json
tree -H . -T 'Docs' -o
index.html docs/
```