

```
#!/usr/bin/env bash
#
# git-list-added.sh – List files with the date they were first added to the Git
# repo
#
# SYNOPSIS
# git-list-added.sh [OPTIONS] [PATH...]
#
# DESCRIPTION
# For each file under version control (in the specified PATHs, or . by default
# t),
# show the date of the commit in which it was introduced.
#
# The output is sorted chronologically (oldest additions first). Dates are in
# ISO 8601 format (YYYY-MM-DDTHH:MM:SS±TZ).
#
# OPTIONS
# -h, --help    display this help and exit
# -r, --reverse sort in reverse order (newest additions first)
#
# REQUIREMENTS
# • git (>= 2.x)
# • bash, coreutils (readlink, sort, printf)
#
# EXAMPLES
# # List all files in the current repo with their add-dates:
# ./git-list-added.sh
#
# # List only files under src/, newest first:
# ./git-list-added.sh -r src
#
# # Install as a global command (make executable and move into $PATH):
# chmod +x git-list-added.sh
# sudo mv git-list-added.sh /usr/local/bin/git-list-added
#
#####
#

set -euo pipefail

# Help text function
show_help() {
    awk 'NR<4 { next } /^#####/'
    #####$/ { exit } { print } "$0"
}

# Parse options
REVERSE_SORT=0
PATHS=()

while (( $# )); do
    case "$1" in
        -h|--help)
            show_help
            exit 0
            ;;
        -r|--reverse)
            REVERSE_SORT=1
            shift
            ;;
        --) # end of options
            shift
            break
            ;;
        -*)
            printf "Unknown option: %s\n" "$1" >&2
            exit 1
            ;;
        *)
            PATHS+=("$1")
            shift
            ;;
    esac
done

# Default to current directory if no paths given
if [ $#PATHS[@] -eq 0 ]; then
    PATHS=(.)
fi

# Ensure we are inside a Git repository
if ! git rev-parse --is-inside-work-tree &>/dev/null; then
    echo "Error: not inside a Git working tree." >&2
    exit 1
fi
```

```
# Collect and print: <date> <tab> <path>
# For each file, find the commit that first introduced it
{
    # List all tracked files under given paths
    git ls-files -- "${PATHS[@]}" |
    while IFS= read -r file; do
        # --diff-filter=A : only commits where the file was Added
        # --format=%cl : committer date, ISO 8601
        date_added=$(git log --diff-filter=A --format=%cl -- "$file" | tail -
n1)
        # If for some reason a file has no history (should not happen), skip it
        if [ -z "$date_added" ]; then
            continue
        fi
        # Print date, tab, path
        printf "%s\t%s\n" "$date_added" "$file"
    done
} | sort -t"$\t" -k1,1$( [ "$REVERSE_SORT" -eq 1 ] && printf 'r' )

exit 0
```
