Glob Patterns Cheatsheet

Overview

Glob patterns are a way to specify sets of filenames with wildcard characters. They are primarily used in shell environments to expand file lists or perform batch operations on multiple files.

Basic Wildcards

- `*`: Matches any number of characters (including zero).
  - Example: `*.txt` -> All files with `.txt` extension (e.g., `notes.txt`, `log.txt`).
  - Example: `img*` -> All files and folders starting with `img` (e.g., `img1.png`, `images/`).
- `?`: Matches exactly one character.
  - Example: `file?.log` -> Matches `file1.log`, `fileA.log`, but NOT `file10.log`.
- `[abc]`: Matches any one character listed inside the brackets.
  - Example: `data[12].csv` -> Matches `data1.csv` or `data2.csv`.
- `[a-z]`: Matches any one character within the specified range.
  - Example: `[0-9].log` -> Sequential numeric single-character file names.
- `[!abc]` or `[^abc]`: Matches any one character NOT listed in the set. (Symbol varies by shell).

Advanced and Recursive Patterns

- `**`: Recursively matches directories and subdirectories. (Supported in modern shells or with specific flags).
  - Example: `**/README.md` -> Finds `README.md` in any directory within the project.
- `{}` (Brace Expansion): Generates combinations of strings separated by commas. (Technically a shell feature but often used with Globs).
  - Example: `mv file.{txt,md}` -> Expands to `mv file.txt file.md`.
  - Example: `src/*.{js,ts}` -> Selects all JS and TS files in the `src` directory.

Common Use Cases

- List all files with specific extensions:
  ```
  ls *.{jpg,png,gif}
  ```
- Search for source code in subdirectories:
  ```
  ls src/**/*.ts
  ```
- Find test files following a numeric pattern:
  ```
  ls test[0-9]*.py
  ```
- Select everything except hidden files:
  ```
  ls *
  ```
- Select all files including hidden ones:
  ```
  ls .* *
  ```

Shell Specifics

Globe behavior can vary depending on the shell (Bash, Zsh, Fish). For example, in Bash, you might need `shopt -s globstar` to enable the `**` pattern. Some shells might require escaping (`\`) special characters in certain contexts.