

Notes 6: Wildcards and Brace Expansion

What Are Wildcards?

Wildcards are special characters used to match filenames or directory names. They help perform actions on multiple files without having to type each name individually.

Wildcard Table

Wildcard	Definition	Example
*	Matches 0 or more characters	<code>ls *.txt</code> lists all <code>.txt</code> files
?	Matches exactly 1 character	<code>ls f?le.txt</code> matches <code>file.txt</code> , <code>flle.txt</code> , etc.
[]	Matches 1 character from a set	<code>ls [ab]*</code> matches files starting with <code>a</code> or <code>b</code>

Usage Examples

* Wildcard (zero or more characters)

- 1. `ls *.sh`
Lists all files ending in `.sh`.
 - 2. `ls My_*`
Lists files starting with `My_`.
 - 3. `ls *program*`
Lists files containing the word `program`.
-

? Wildcard (exactly one character)

- 1. `ls file?.txt`
Matches `file1.txt`, `file2.txt`, but not `file10.txt`.
 - 2. `ls ?.sh`
Matches files like `a.sh`, `b.sh`, but not `ab.sh`.
 - 3. `ls a?.txt`
Matches `a1.txt`, `ab.txt`, etc.
-

[] Wildcard (character from a set or class)

- 1. `ls [A-Z]*`
Lists files that start with an uppercase letter.

- 2. `ls *[0-9]*`
Lists files containing numbers.
- 3. `ls *[:punct:]*`
Lists files containing punctuation characters.

POSIX Character Classes (inside `[]`)

POSIX Class	Equivalent	Matches	Example
<code>[:upper:]</code>	<code>[A-Z]</code>	Uppercase letters	<code>ls *[:upper:]*</code>
<code>[:lower:]</code>	<code>[a-z]</code>	Lowercase letters	<code>ls *[:lower:]*</code>
<code>[:digit:]</code>	<code>[0-9]</code>	Digits	<code>ls *[:digit:]*</code>
<code>[:alpha:]</code>	<code>[A-Za-z]</code>	Alphabetic characters	<code>ls *[:alpha:]*</code>
<code>[:alnum:]</code>	<code>[A-Za-z0-9]</code>	Alphanumeric characters	<code>ls *[:alnum:]*</code>
<code>[:punct:]</code>	punctuation	Punctuation marks	<code>ls *[:punct:]*</code>

Brace Expansion `{ }`

Brace expansion is used to generate arbitrary strings.

Examples:

- 1. `echo file{1,2,3}.txt`
Expands to: `file1.txt file2.txt file3.txt`
- 2. `mkdir project_{A,B,C}`
Creates: `project_A,project_B,project_C`
- 3. `cp file{A,B}.txt`
Copies `fileA.txt` and `fileB.txt`

When Should I Use Wildcards?

I can use wildcards when I want to work with many files at once:

- Copy all `.jpg` images to a flash drive:
`cp *.jpg /media/usb`
- Delete all temporary files:
`rm *.tmp`
- List all files that include a date:
`ls *2024*`