

Notes 4: Wildcards

1. Asterisk (*):

Usage:

The asterisk (*) wildcard matches any sequence of characters (including zero characters) within a filename.

Example:

Suppose you have files named `file1.txt`, `file2.txt`, and `file3.txt`. To list all files that end with `.txt`, you can use `*.txt`, which will match any filename that ends with `.txt`.

2. Question Mark (?):

Usage:

The question mark (?) wildcard matches any single character within a filename.

Example:

Suppose you have files named `file1.txt`, `file2.txt`, and `file3.txt`. To list files where the second character is a digit, you can use `file?.txt`, which will match any filename with file followed by any single character and then `.txt`.

3. Square Brackets ([]):

Usage:

Square brackets ([]), also known as character classes, match any single character specified within the brackets.

Example

Suppose you have files named `file1.txt`, `file2.txt`, and `file3.txt`. To list files with names containing either `file1.txt` or `file2.txt`, you can use `file[12].txt`, which will match any filename that starts with file, followed by either 1 or 2, and ends with `.txt`.

Usage:

Brace expansion, denoted by curly braces {}, allows you to generate multiple strings or filenames based on a pattern or a list of values separated by commas.

Example:

Suppose you want to create files named `file1.txt`, `file2.txt`, and `file3.txt`. Instead of creating each file individually, you can use brace expansion: `touch file{1..3}.txt`. This command will create files `file1.txt`, `file2.txt`, and `file3.txt`.

