Understanding Wildcards and Brace Expansion in Shell

Wildcards

Wildcards are symbols that enable pattern matching in file and directory names. The most commonly used wildcards are `*`, `?`, and `[]`.

Asterisk (`*`)

The `*` wildcard matches zero or more characters in a filename or directory.

- **Example**: `ls *.txt` lists all files in the current directory with a `.txt` extension.

Question Mark (`?`)

The `?` wildcard matches exactly one character.

- **Example**: `ls ?.txt` lists all files in the current directory that have one character followed by `.txt`.

Square Brackets (`[]`)

The `[]` wildcard matches any one of the characters enclosed within the brackets.

- **Example**: `ls [a-e].txt` lists files named `a.txt`, `b.txt`, `c.txt`, `d.txt`, and `e.txt` in the directory.

- You can also use ranges: `ls [1-3].txt` would match `1.txt`, `2.txt`, and `3.txt`.

Brace Expansion (`{}`)

Brace expansion is used to generate arbitrary strings. It is particularly useful for creating sequences of files or directories.

- **Syntax**: `{item1,item2,item3}`

Single Item Expansion

- **Example**: `mkdir {2020,2021,2022}_Report` creates directories named `2020_Report`, `2021_Report`, and `2022_Report`.

Sequence Expansion

- **Example**: `touch report_{1..5}.txt` creates files named `report_1.txt`, `report_2.txt`, `report_3.txt`, `report_4.txt`, and `report_5.txt`.
- You can also specify a step value: `echo {0..10..2}` prints even numbers from 0 to 10.

Nested Expansion

Brace expansion can be nested to create combinations of expansions.

- **Example**: `mkdir {A,B}_{1,2,3}` creates directories `A_1`, `A_2`, `A_3`, `B_1`, `B_2`, and `B_3`.