Notes 7

The * wildcard

Definition

Represents any sequence or number of characters, from none to infinity. Used in combination with regular characters for more precise wildcard filtering.

Examples

All of an extension type

```
ls *.pdf # List all pdf files in directory
pricesexport.pdf
finances.pdf
q2_report.pdf
```

Pattern matching

```
mv price*.txt ~/Documents/
```

> Move any file containing 'price', after price any number of any character, and matching the extension `.txt`

```
ls -hS *backup.*
```

> Matches any file with `backup.` , allowing for any extension to be matched. Also includes -h (human readable) and -S (sort by size, largest first)

The ? wildcard

Definition

Represents a single unknown character. Purpose is to match specific amount of characters, as opposed to `*` matching any number of characters.

Examples

Extension matching

```
ls -a *.mp?
```

> This will match any file's name with the extension matching `mp` plus a single character.

Hidden file matching

```
cp .??*.??? /.secrets
```

> This command will copy all hidden files (starting with a `.`) to the `.secrets` directory.
> Each must have at least two characters, as indicated by the double `?`.
> Each must have an extension exactly three characters long.

Partial names

```
rm file_???.txt
```

> Removes any file with exactly three characters between `_` and `.`.

The [] wildcard

Definition

Matches to a given set of characters.

Use `!` at the start to give the opposite of the given set. You may use predefined sets of characters, also known as POSIX classes, for convinience.

Examples

Letter matching

```
ls class-[a-e].txt
```

> Makes a list of every .txt file containing class-a through class-e.

Combining wildcards

```
ls CoD-*-[1-5].zip
```

> Would match CoD, then any number of characters, then any number between one and five.

Brace Expansion

Usage

Brace expansion is used when needing create multiple strings of text at once, such as batch creating, modifying, or viewing files or directories.

Examples

Creating directory structure

```
mkdir -p ~/projects/automation-
project/{scripts/{legacy,indev},cli,data_processing,logs}
```

Can use nested braces in order to create more complex directory structures.

Listing multiple directories

```
ls -lah ~/.config/{gtk-4.0,KDE,xfce4}
```

Lists the contents of all three directories, with labels for each directories' contents.