Glob Patterns and Wildcards: Takeaways

Syntax

- Wildcards:
  - `?` matches any single character.
  - `*` matches any string of characters.
  - `[list_of_characters]` matches any characters in `list_of_characters`.
  - `[!list_of_characters]` matches any characters **not** in `list_of_characters`.
  - `[[:alpha:]]` matches any letter.
  - `[[:digit:]]` matches any number.
  - `[[:alnum:]]` matches any letter or number.
  - `[[:lower:]]` matches any lowercase letter.
  - `[[:upper:]]` matches any uppercase letter.

Concepts

- We can use **wildcards** to create patterns to match groups of filenames.
- These patterns, called **glob patterns**, work in a similar way to regular expressions, albeit with different rules.
- We can use glob patterns with most commands, making them an extremely powerful tool.
- Because they're very powerful, we need to be careful with them, especially when it comes to commands that modify the filesystem (like `rm`).

Resources

- Character classes in GNU.
- Globbing and Regex: So Similar, So Different.
- Glob patterns and regular expressions summary.
- The glob function.
- Locale.
- `find` :
    - How to Find a File in Linux Using the Command Line
    - 35 Practical Examples of Linux `find` Command
    - Unix Find Tutorial
- The `locate` command — an alternative to `find` :
    - Linux `locate` command
    - 10 Useful `locate` Command Practical Examples for Linux Newbies