Notes 7 - Wildcards

The * (star/asterisk) wildcard

Definition:

- The asterisk (*) matches zero or more characters in a filename.

Examples:

- List all the files in a given directory
  - `ls Downloads/*`
- List all the text files in a given directory
  - `ls Downloads/*.txt`

The ? (question mark) wildcard

Definition:

- The ? wildcard metacharacter matches *precisely one character*
- The ? wildcard proves very useful when working with hidden files(dot files).
  - To list all hidden files use: ls .??* which will match all files that starts with a . or .. and have any charachter after it.

Examples:

- List all the hidden files in the current working directory - `ls ./. ??`
- list all the files that have 2 character in the filename between letters b and k
  - `ls b??k`

The [] (square brackets) wildcard

Definition:

- The brackets wildcard matches a single character in a range,
- The brackets wildcard use the exclamation mark to reverse the match
  - For example: match everything except wowels[!aelou] or any charachter except numbers [!0-9]

Examples:

- To match all files that have a vowel after letter f:
  - `ls f[aeiou]*`
- To match all files that do not have a vowel after letter f:
  - `ls f[!aeiou]*`
- To match all files that have a range of letters after f::
  - `ls f[a-z]*`

Brace Expansion

Definition:

- Brace expansion is not a wildcard but a feature of the bash shell that allows you to create strings without needing loops. The strings can be filenames, sequences, or patterns. Brace expansion is handled before file globing and variable expansion.
- Brace expansion is used in the following manner:
  - Open brace ({) + pattern separated by commas with no spaces + closing brace (}).

Examples:

- Create 3 html files
  - `touch {index, about, contratcs}.html`
- Remove the same 100 files created in the previous example:
  - `rm file{A..Z}.txt`
- Create 10 files numbered 1 to 10
  - `touch file{A..Z}.txt`