# Note 6

## The * Wildcard

```
It can represent any number of characters, including none.
```

## Examples

```
1. List all txt and python files
    ->>  ls -A *.txt *.py
2. List all the files that have 'demo' in the name
    ->>  ls -A *demo*
3. Move all the files inside a directory
    ->>  mv Pictures/* ~/Backup/
```

## The ? Wildcard

```
Useful for targeting files with single-character differences.
```

### Examples

```
1. Copy all the files that have 2 characters between  letters.
    ->>  cp Downloads/b?k.pdf Documents/
2. List all the files with a 2 letters file extension
    ->>  ls -A Scripts/*.?? Programs/program.?? Downloads/setup*.??
3. Remove all the hidden files in a given directory
    ->>  rm Documents/.??*.doc
```

## The [] Wildcard

```
You can define a set or a range of characters.
```

### Examples

```
1. List all the text files that start with an uppercase letter and all the
python files that start with a number.
    ->>  ls -A[A-Z]*.txt [0-9]*.py
2. List all the ruby file that do not start a number.
```

```
    ->>  ls -A[!0-9]*.rb
3. List all the files that have one of the characters in a set before the
extension.
    ->>  list *[xyz].*
```

## Brace Expansion {}

```
The brace expansion {} allows you to automatically generate a series of
text strings from a pattern. It doesn't search for files like wildcards (*,
?, []), but instead creates multiple words or commands before executing
them.
```

Examples:

```
1. The braces {1,2,3} are expanded in several versions of the word fileThe
braces
    {1,2,3} are expanded in several versions of the word file.
    ->>  echo file{1,2,3}.txt
2. We use {1..5} to generate a sequence from 1 to 5.
    ->>  echo file{1..5}.txt
3. This generates files with letters from A to C.
    ->> echo file{A..C}.txt
```