

Wild Cards

* - represents zero or more characters

? - represents a single character

[] - represents a range of characters

* (asterisk) Wild Card

- * represents zero or more characters

Examples:

- the below wild card expression matches all the files (not hidden) in the current directory

```
# ls *
```

```
output: b ball.txt blah.txt bob test.txt
```

- The below wild card expression matches the file which starts with first character as b followed by zero or any number of characters.

```
#ls b*
```

```
output: b ball.txt blah.txt bob
```

- The below wild card expression matches the files which start with zero or any number of characters followed by .txt

```
# ls *.txt
```

```
output: ball.txt blah.txt test.txt
```

? Wild card

? - represents a single character

Examples:

- The below wildcard matches the files which starts with zero or any number of characters and ends with three characters followed by a dot(.)

```
#ls *.???
```

output:

ball.txt blah.txt test.txt

- The below wildcard matches the files which starts with 4 characters followed by a dot(.) and then followed by 3 characters.

```
#ls ?????.???
```

output:

ball.txt blah.txt test.txt

[] Wild Card

[] - represents a range of characters

The following expression provides all the objects in the current directory which starts with any of the in a range of a through f.

```
#ls [a-f]*
```

output:

```
b ball blah.txt bob
```

Summary:

Wildcards

* - represents zero or more characters

? - represents a single character

[] - represents a range of characters

❑ You can use wildcard combined

```
$ls ???.*
```

```
$ls *.???[a-h]
```

❑ Ranges can be specified using digit class

```
$ls *.???[[:digit:]]
```

❑ Wildcard can be used with command like ls, mv, cp, rm etc..