Notes 6

POSIX class	Represents	Means	Example using `ls`
`[:upper:]`	`[A-Z]`	Uppercase letters	`ls *[[:upper:]]*` - List files with uppercase letters
`[:lower:]`	`[a-z]`	Lowercase letters	`ls *[[:lower:]]*` - List files with lowercase letters
`[:digit:]`	`[0-9]`	Digits	`ls *[[:digit:]]*` - List files containing digits
`[:alpha:]`	`[A-Za-z]`	Alphabetic characters	`ls *[[:alpha:]]*` - List files with alphabetic characters
`[:alnum:]`	`[A-Za-z0-9]`	Alphanumeric characters	`ls *[[:alnum:]]*` - List files with alphanumeric characters
`[:space:]`	`[ \t\n\r\f\v]`	Whitespace characters	`ls *[[:space:]]*` - List files with spaces in names
`[:punct:]`	Punctuation	Punctuation characters	`ls *[[:punct:]]*` - List files with punctuation characters
`[:blank:]`	`[ \t]`	Space and tab	`ls *[[:blank:]]*` - List files with spaces or tabs
`[:xdigit:]`	`[0-9A-Fa-f]`	Hexadecimal digits	`ls *[[:xdigit:]]*` - List files with hexadecimal characters
`[:cntrl:]`	Control characters	Control characters	`ls *[[:cntrl:]]*` - List files with control characters
`[:print:]`	Printable	Printable characters	`ls *[[:print:]]*` - List files with printable characters
`[:graph:]`	Graphical	Visible characters (not spaces)	`ls *[[:graph:]]*` - List files with graphical characters
`[:word:]`	`[A-Za-z0-9_]`	Word characters (alphanumeric + underscore)	`ls *[[:word:]]*` - List files containing word characters
`[:ascii:]`	ASCII characters	All ASCII characters (0-127)	`ls *[[:ascii:]]*` - List files containing ASCII characters

* Wildcard

```
* matches 0 to any number of characters
```

Example:

```
* ls -1d wildcard_extra_practice/*/

* mv wildcard_extra_practice/*.sh wildcard_extra_practice/scripts/

* mkdir wildcard_extra_practice/images/
```

? Wildcard

```
* matches 1 character
```

Example:

```
*  ls -1X wildcard_extra_practice/*.????

* ls -1X wildcard_extra_practice/i*.????

* ls -1X wildcard_extra_practice/*.???x
```

[] WIldcard

```
* matches 1 character from a set
```

Example:

```
* ls *[[:upper:]]*
* ls *[[:lower:]]*
* ls *[[:digit:]]*
* ls f[a-z]*
```

Brace Expansion

```
* allows you to generate aribiturary strings to use with commands
```

Example:

```
* mkdir -p music/{jazz,rock}/{mp3files,videos,oggfiles}/new{1..3}

* touch webdite{1..5}.html

*rm -r {dir1,dir2,dir3,file.txt.file.py}
```