

Bash Arguments Shortcuts

1. Access the last argument of the previous command - `!$`
- `!$`
2. Reuse the entire previous command - `!!`
- `!!`
3. Use `Alt + .` (dot) to insert the last argument interactively

```
cat <Alt + .>
```

Other Bash Shortcuts

1. Print the Current PID of the current shell

```
echo $$
```
2. Print the **PID** of the most recently executed background job

```
echo $!
```
2. Command History Shortcuts
 - `!n` → Run command number **n** from history
`!25`
 - `!string` → Run the most recent command **starting with string**
`!echo`
 - Runs the last command that started with `echo`.

Wildcards (Globbing)

In Linux (globbing patterns), wildcards are special characters that help you match and select files and directories using a pattern.

1. ***** (**Asterisk**) – Matches any number of characters (including none).

```
ls *.txt
```

```
rm file*
```
2. **?** (**Question Mark**) – Matches exactly one character.

```
ls file?.txt
```
3. **[]** (**Square Brackets**) – Matches any one character inside the brackets.

```
ls file[1-4].txt
```
4. **{ }** (**Brace Expansion**) – Expands into multiple values.

```
touch file{1..4}.txt
mkdir {work,personal,projects}
```

5. **!** or **^** inside **[]** – Matches anything **except** the given characters.

```
ls file[!0-9].txt
```

Advanced Expansion Tricks

6. Brace Expansion with Step Size

```
echo {1..10..2}
```

7. Expanding Letters

```
echo {a..f}
```

8. Using **\$(command)** to Insert Output into a Command
bash touch file_\$(date +%Y-%m-%d).txt

Where to store scripts

Best Practices

- Personal scripts: `~/bin/`
- System-wide scripts: `/usr/local/bin/`
- Admin scripts: `/usr/local/sbin/`