# Prefixing

## Prefix Styles

concurrently will by default prefix each command's outputs with a zero-based index, wrapped in square brackets:

$ concurrently "echo Hello there" "echo 'General Kenobi!'"

[0] Hello there

[1] General Kenobi!

[0] echo Hello there exited with code 0

[1] echo 'General Kenobi!' exited with code 0

If you've given the commands names, they are used instead:

$ concurrently --names one,two "echo Hello there" "echo 'General Kenobi!'"

[one] Hello there

[two] General Kenobi!

[one] echo Hello there exited with code 0

[two] echo 'General Kenobi!' exited with code 0

There are other prefix styles available too:

| **Style** | **Description** |
| --- | --- |
| index | Zero-based command's index |
| name | The command's name |
| command | The command's line |
| time | Time of output |
| pid | ID of the command's process (PID) |
| none | No prefix |

Any of these can be used by setting the --prefix/-p flag. For example:

$ concurrently --prefix pid "echo Hello there" "echo 'General Kenobi!'"

[2222] Hello there

[2223] General Kenobi!

[2222] echo Hello there exited with code 0

[2223] echo 'General Kenobi!' exited with code 0

It's also possible to have a prefix based on a template. Any of the styles listed above can be used by wrapping it in {}. Doing so will also remove the square brackets:

$ concurrently --prefix "{index}-{pid}" "echo Hello there" "echo 'General Kenobi!'"

0-2222 Hello there

1-2223 General Kenobi!

0-2222 echo Hello there exited with code 0

1-2223 echo 'General Kenobi!' exited with code 0

## Prefix Colors

By default, there are no colors applied to concurrently prefixes, and they just use whatever the terminal's defaults are.

This can be changed by using the --prefix-colors/-c flag, which takes a comma-separated list of colors to use.

The available values are color names (e.g. `green`, `magenta`, `gray`, etc), a hex value (such as `#23de43`), or `auto`, to automatically select a color.

$ concurrently -c red,blue "echo Hello there" "echo 'General Kenobi!'"

* List of available color names black
* blue
* cyan
* green
* gray
* magenta
* red
* white
* yellow

Colors can take modifiers too. Several can be applied at once by prepending .<modifier 1>.<modifier 2> and so on.

$ concurrently -c red,bold.blue.dim "echo Hello there" "echo 'General Kenobi!'"

* List of available modifiers reset
* bold
* dim
* hidden
* inverse
* italic
* strikethrough
* underline

A background color can be set in a similary fashion.

$ concurrently -c bgGray,red.bgBlack "echo Hello there" "echo 'General Kenobi!'"

* List of available background color names bgBlack
* bgBlue
* bgCyan
* bgGreen
* bgGray
* bgMagenta
* bgRed
* bgWhite
* bgYellow

## Prefix Length

When using the command prefix style, it's possible that it'll be too long.

It can be limited by setting the `--prefix-length`/`-l` flag:

$ concurrently -p command -l 10 "echo Hello there" "echo 'General Kenobi!'"

[echo..here] Hello there

[echo..bi!'] General Kenobi!

[echo..here] echo Hello there exited with code 0

[echo..bi!'] echo 'General Kenobi!' exited with code 0

It's also possible that some prefixes are too short, and you want all of them to have the same length.

This can be done by setting the `--pad-prefix` flag:

$ concurrently -n foo,barbaz --pad-prefix "echo Hello there" "echo 'General Kenobi!'"

[foo ] Hello there

[foo ] echo Hello there exited with code 0

[barbaz] General Kenobi!

[barbaz] echo 'General Kenobi!' exited with code 0

[!NOTE] If using the pid prefix style in combination with [--restart-tries](http://./restarting.md), the length of the PID might grow, in which case all subsequent lines will match the new length.

This might happen, for example, if the PID was 99 and it's now 100.