# Redirection and Pipelines: Takeaways ⮦

## Syntax

- Printing `Hello, world!` to the screen: `echo "Hello, world!"`.

- Creating empty files: `touch file1 file2`.

- Creating a file `output` by redirecting the output `cmd`: `cmd >output`.

- Appending the output of running `cmd` to the file `output`: `cmd >>output`.

- Connecting the output of `cmd1` to the input of `cmd2`: `cmd1 | cmd2`.

- Discarding the output of `cmd`: `cmd >/dev/null`.

## Concepts

- Creating empty text files is sometimes useful.
- With output redirection we can save our work into files.
- We should be mindful of situations in which we'll want to append output.
- Redirecting errors isn't as simple as it seems.
- Pipelines allows us to very easily combine commands together.
- A couple of the benefits of pipelines are efficiency and ease of use.
- The Unix Philosophy:

  > This is the Unix philosophy. Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.

## Resources

- The `printf` command
- GNU Project's documentation on Pipelines

- [POSIX standards on pipelines](#)

- The [null device](#)

- [Professor Brian Kernighan](#) on Computerphile [talking about Unix](#)