What is msg?

`msg` is a simple script I use to produce nice and consistent messages, prefixed properly and what not, most of the looks are inspired from pacman and yay

Here's an example of how the messages can look:

```
 $ msg --test

 Status message!
> Other message!
:: Action message!
Error: Error message!
==> Action Warn message!
==> Action Info message!
Success: Success message!
Warning: Warning message!
 -> Notice message!
(1/5) Progress message!
```

All of them having customizable colors and prefixes, meaning if you don't want the error message to say have "Error:" as it's prefix you can change it (At least you'll be able to soon…)

The main objective of this was instead of creating functions inside my scripts for making good looking output I made this script to unify it.

Outside of that you can also color the actual message with escape sequences. Meaning if you want part of the message red you can add `\e[31m` somewhere and to stop the read add `\e[0m` so `msg "An error occured in \e[31mscript\e[0m fix it" error` would produce and error message where the word "script" is in red. For more info on this read up on terminal codes, here's a good place to get a good grip on how it works.

Options

As of right now the decision is to not have any command line arguments outside of `--test` which prints the example message shown above. Which also means all the customization won't be done with command line arguments. Instead it'll be done with environment variables.

That way scripts unless they're evil won't be able to overwrite it as well.

However an issue arises if the user wants (as an example) their warning msg prefix to be "WARNING!!" instead of "Warning:" and a script needs to provide a different message that isn't an warning message but still needs to be in yellow. Lets say the script provides "Issue:" as the third argument, that'll change the "Warning:" to "Issue:"

But as you can guess this would mean when an error comes up it says "WARNING!!" but when this script uses it's own method it'll show just "Issue:" and it might seem out of place.

Perhaps the best idea is just to encourage developers not to provide their own prefix.

How to install!

Currently there's no package on any package manager as I haven't bothered to make one, but I will sometime in the future probably. For now you'll have to copy the `msg` file from the repository and put it in a directory that's in your `$PATH`

A way of doing this that'll work for most people is by entering this command:

```
 curl
https://raw.githubusercontent.com/0neGuyDev/msg/main
/msg -o ~/.local/bin/msg | chmod +x ~/.local/bin/msg
```

However make sure `~/.local/bin/msg` is actually in your `$PATH` not to mention that it exists.