

WP_CLI\Utils\isPiped()



- [Usage](#)
- [Notes](#)
- [Related](#)

Checks whether the output of the current script is a TTY or a pipe / redirect

Usage

```
WP_CLI\Utils\isPiped()
```

@return (bool)

Notes

Returns `true` if `STDOUT` output is being redirected to a pipe or a file; `false` if output is being sent directly to the terminal.

If an env variable `SHELL_PIPE` exists, the returned result depends on its value. Strings like `1`, `0`, `yes`, `no`, that validate to booleans are accepted.

To enable ASCII formatting even when the shell is piped, use the ENV variable `SHELL_PIPE=0`.

```
SHELL_PIPE=0 wp plugin list | cat
```

Note that the `db` command forwards to the `mysql` client, which is unaware of the env variable. For `db` commands, pass the `--table` option instead.

```
wp db query --table "SELECT 1" | cat
```

Internal API documentation is generated from the WP-CLI codebase on every release. To suggest improvements, please submit a pull request.

Related

- [WP_CLI\Utils\write_csv\(\)](#) - Write data as CSV to a given file.
- [WP_CLI\Utils\http_request\(\)](#) - Make a HTTP request to a remote URL.
- [WP_CLI\Utils\get_named_sem_ver\(\)](#) - Compare two version strings to get the named semantic version.
- [WP_CLI\Utils\parse_ssh_url\(\)](#) - Parse a SSH url for its host, port, and path.
- [WP_CLI\Utils\basename\(\)](#) - Locale-independent version of basename()
- [WP_CLI\Utils\proc_open_compat\(\)](#) - Windows compatible ``proc_open()``. Works around bug in PHP, and also deals with *nix-like ``ENV_VAR=blah cmd`` environment variable prefixes.
- [WP_CLI\Utils\esc_like\(\)](#) - First half of escaping for LIKE special characters % and _ before preparing for MySQL.

📄 #1

📄 cocobook-sync 📄 2026-10-28 11:41:12 UTC
📄 cocobook-sync 📄 2026-10-28 11:41:12 UTC