# Contributing guidelines

Here are a couple things to take into account when contributing to swizzin.

## Directories and files

- Please use `/opt` to install any new software. \*\*Avoid using home directories to install application binaries/source\*\*

- When creating configuration for users, please add it to an appropriate folder under `~/.config/` when possible

- Please make sure to put any necessary code output into the default log files for swizzin, (i.e. `swizzin.log` and `install.log` under `/root/logs`) \*including\* the `stderr`

- Make sure to create `.lock` files for your application under `/install/`

- Record any application information that might need to be dynamically retrieved into the .lock file above.

## Bash code styleguide

### Code re-use

Please familiarise yourself with the functions available under `sources/functions` as they handle a large amount of steps you might need to do manually.

Whenever you are contributing code that might be generalised and useful for other applications, please add the functions to this location. When doing so, please give a quick comment into the file on what the purpose and possible return output is. e.g:

```bash

# Retrieves all users that are managed by swizzin

# Returns usernames separated by newline

```

### Shellcheck

Please use `shellcheck` and resolve any warnings for the code you have contributed.

### Codename-based logic

When making logic based on distribution codenames, please structure it in such a way that introduces as little need for maintenance in the future as possible.

A practical example for this is handling packages that are no longer packaged for newer LTS releases. In such scenarios, please make it so that the "default" behaviour is for newer LTS releases, and the older LTS releases are treated as the outliers. e.g.:

```bash

if [[ $codename =~ ("xenial"|"stretch") ]]; then

mcrypt=php-mcrypt

else

mcrypt=

fi

# Apt install line which has $mcrypt in the package lsit

```

In this example, the default (`else`) behaviour triggers for current releases, and only the old ones have the modified behaviour

### Return codes

Please use `exit 1` or `return 1` when handling exiting due to some run-time errors

## Python applications

As a principle, please avoid installing Python2 applications.

Due to how Python is being versioned, please see the python functions available in `sources/function/python`.

Please use virtual environments only for applications that require a specific or an outdated version of Python.

## Service files

Please include service files for application management. Use the `%i` feature for multi-user applications, avoid for single-instance applications.

## User management

Whenever possible, please describe how is user management done within your application, and how you have adjusted `box` to handle this.

## Documentation

For new applications, please make sure to make the necessary documentation pages on the [Swizzin Docs](https://github.com/liaralabs/docs.swizzin.ltd) repo. It's nice and convenient when you include the link to the Documentation repo PR in the PR here.