Contributing to JuNest

=====================

First off, thanks for taking the time to contribute!

The following is a set of guidelines for contributing to JuNest.

These are just guidelines, not rules, use your best judgment and

feel free to propose changes to this document in a pull request.

\*\*Table of Contents\*\*

- [How Can I Contribute?](#how-can-i-contribute)

- [Reporting Bugs](#reporting-bugs)

- [Suggesting Enhancements](#suggesting-enhancements)

- [Your First Code Contribution](#your-first-code-contribution)

- [Styleguides](#styleguides)

- [Git Commit Messages](#git-commit-messages)

- [Documentation Styleguide](#documentation-styleguide)

- [Shell Styleguide](#shell-styleguide)

- [Versioning](#versioning)

## How Can I Contribute? ##

### Reporting Bugs ###

This section guides you through submitting a bug report for JuNest.

#### Before submitting a bug report ####

You might be able to find the cause of the problem and fix things yourself.

- \*\*Check the [troubleshooting section](https://github.com/fsquillace/junest#troubleshooting)\*\*

- \*\*Check if you can reproduce the problem with the latest version of JuNest\*\*

- \*\*Check for [existing open/closed issues](https://github.com/fsquillace/junest/issues?utf8=%E2%9C%93&q=is%3Aissue)\*\*

- If the bug has already been suggested, add a comment to the existing issue instead of opening a new one.

#### How Do I Submit A (Good) Bug Report? ####

Bugs are tracked as [GitHub issues](https://guides.github.com/features/issues/) in the [JuNest issues page](https://github.com/fsquillace/junest/issues).

Explain the problem and include additional details to help maintainers reproduce the problem:

\* \*\*Use a clear and descriptive title\*\* for the issue to identify the problem.

\* \*\*Describe the exact steps which reproduce the problem\*\* in as many details as possible. For example, start by explaining how you started JuNest, e.g. which command exactly you used in the terminal. When listing steps, \*\*don't just say what you did, but explain how you did it\*\*. For example.

\* \*\*Provide specific examples to demonstrate the steps\*\*. Include links to files or GitHub projects, or copy/pasteable snippets, which you use in those examples. If you're providing snippets in the issue, use [Markdown code blocks](https://help.github.com/articles/markdown-basics/#multiple-lines).

\* \*\*Describe the behavior you observed after following the steps\*\* and point out what exactly is the problem with that behavior.

\* \*\*Explain which behavior you expected to see instead and why.\*\*

\* \*\*Put the bug label to the issue.\*\*

Include details about your configuration and environment:

\* \*\*Which version of JuNest are you using?\*\*

\* \*\*What's the name and version of the OS you're using\*\*?

\* \*\*Are you running JuNest in a virtual machine?\*\* If so, which VM software are you using and which operating systems and versions are used for the host and the guest?

\* \*\*Which packages do you have installed?\*\* You can get that list by running `pacman -Qq`.

#### Template For Submitting Bug Reports ####

[Short description of problem here]

\*\*Reproduction Steps:\*\*

1. [First Step]

2. [Second Step]

3. [Other Steps...]

\*\*Expected behavior:\*\*

[Describe expected behavior here]

\*\*Observed behavior:\*\*

[Describe observed behavior here]

\*\*JuNest version:\*\* [Enter JuNest version here]

\*\*OS and version:\*\* [Enter OS name and version here]

\*\*Installed packages:\*\*

[List of installed packages here]

\*\*Additional information:\*\*

\* Problem started happening recently, didn't happen in an older version of JuNest: [Yes/No]

\* Problem can be reliably reproduced, doesn't happen randomly: [Yes/No]

### Suggesting Enhancements ###

This section guides you through submitting an enhancement suggestion for JuNest, including completely new features and minor improvements to existing functionality.

#### Before Submitting An Enhancement Suggestion ####

\* \*\*Check if you're using the latest version of JuNest\*\*

- \*\*Check for [existing open/closed issues](https://github.com/fsquillace/junest/issues?utf8=%E2%9C%93&q=is%3Aissue)\*\*

- If enhancement has already been suggested, add a comment to the existing issue instead of opening a new one.

#### How Do I Submit A (Good) Enhancement Suggestion? ####

Enhancement suggestions are tracked as [GitHub issues](https://guides.github.com/features/issues/) in the [JuNest issues page](https://github.com/fsquillace/junest/issues).

Create an issue on that repository and provide the following information:

\* \*\*Use a clear and descriptive title\*\* for the issue to identify the suggestion.

\* \*\*Provide a step-by-step description of the suggested enhancement\*\* in as many details as possible.

\* \*\*Provide specific examples to demonstrate the steps\*\*. Include copy/pasteable snippets which you use in those examples, as [Markdown code blocks](https://help.github.com/articles/markdown-basics/#multiple-lines).

\* \*\*Describe the current behavior\*\* and \*\*explain which behavior you expected to see instead\*\* and why.

\* \*\*Specify which version of JuNest you're using.\*\*

\* \*\*Specify the name and version of the OS you're using.\*\*

\* \*\*Put the enanchement label to the issue.\*\*

#### Template For Submitting Enhancement Suggestions ####

[Short description of suggestion]

\*\*Steps which explain the enhancement\*\*

1. [First Step]

2. [Second Step]

3. [Other Steps...]

\*\*Current and suggested behavior\*\*

[Describe current and suggested behavior here]

\*\*Why would the enhancement be useful to most users\*\*

[Explain why the enhancement would be useful to most users]

[List some other text editors or applications where this enhancement exists]

\*\*JuNest Version:\*\* [Enter JuNest version here]

\*\*OS and Version:\*\* [Enter OS name and version here]

### Your First Code Contribution ###

All JuNest issues are tracked as [GitHub issues](https://guides.github.com/features/issues/) in the [JuNest issues page](https://github.com/fsquillace/junest/issues).

#### Pull Requests ####

\* Fork the repo and create your feature branch from \*\*\*dev\*\*\*.

\* If you make significant changes, please add tests too.

Get familiar with [shunit](https://github.com/kward/shunit2).

\* If you've changed APIs, please update the documentation

\* Follow the [Shell styleguide](#shell-styleguide).

\* Document new code based on the

[Documentation Styleguide](#documentation-styleguide).

\* End files with a newline.

\* Follow the [Git commit messages](#git-commit-messages).

\* Send a [GitHub Pull Request to JuNest](https://github.com/fsquillace/junest/compare/dev...) with a clear list of what you've done (read more about [pull requests](http://help.github.com/pull-requests/)).

\* Put \*\*dev as the base branch\*\* and NOT the master one.

#### Unit Tests ####

To run unit tests:

```sh

./tests/unit-tests/unit-tests.sh

```

#### Integration Tests ####

Generally, there is no need to run integration tests locally

since [Travis](https://travis-ci.org/fsquillace/junest) will run as

soon as the pull request gets created.

## Styleguides ##

### Git Commit Messages ###

\* Follow the [seven rules](http://chris.beams.io/posts/git-commit/#seven-rules) of a great Git commit message

\* Reference issues and pull requests liberally

\* Consider starting the commit message with an applicable emoji:

\* :art: `:art:` when improving the format/structure of the code

\* :racehorse: `:racehorse:` when improving performance

\* :non-potable\_water: `:non-potable\_water:` when plugging memory leaks

\* :memo: `:memo:` when writing docs

\* :penguin: `:penguin:` when fixing something on Linux

\* :apple: `:apple:` when fixing something on Mac OS

\* :checkered\_flag: `:checkered\_flag:` when fixing something on Windows

\* :bug: `:bug:` when fixing a bug

\* :fire: `:fire:` when removing code or files

\* :green\_heart: `:green\_heart:` when fixing the CI build

\* :white\_check\_mark: `:white\_check\_mark:` when adding tests

\* :lock: `:lock:` when dealing with security

\* :arrow\_up: `:arrow\_up:` when upgrading dependencies

\* :arrow\_down: `:arrow\_down:` when downgrading dependencies

\* :shirt: `:shirt:` when removing linter warnings

\* :package: `:package:` when bumping the version

### Documentation Styleguide ###

\* Use [Markdown](https://daringfireball.net/projects/markdown).

### Shell Styleguide ###

\* Use [google shell styleguide](https://google.github.io/styleguide/shell.xml)

#### Function documentation ####

For function documentation follows the example below:

#######################################

# Cleanup files from the backup dir.

#

# Globals:

# VAR1 (RO,bool) : `my\_func` access to VAR1.

# VAR2 (WO) : `my\_func` change the value of VAR2.

# VAR3 (RW) : `my\_func` read and write to VAR3.

# Arguments:

# arg1 ($1,int) : Directory to cleanup.

# arg2 ($2-) : Command to execute for the cleanup.

# Returns:

# 0 : Cleanup completed successfully.

# 101 : Backup directory is not readable.

# $NOT\_DIR\_ERROR : Backup directory is not a directory.

# Output:

# None

#######################################

my\_func() {

local arg1=$1

shift

local arg2=$@

...

}

The documentation is divided by a description of the function, a `Globals`,

`Arguments`, `Returns` and `Output` sections. If a section does not need details

use the word `None` inside of it.

`Globals` section provides all global variables that interact with the function.

`Arguments` section provides the list of arguments to pass to the function. Use

the parenthesis to indicate the position of the arguments:

- `$1` : Argument is in position one.

- `$2-` : Argument takes all args from position two up to the end.

- `$@` : Argument takes all args.

- `$3?` : Argument is optional.

Variables defined in `Globals` and `Arguments` sections can have the following

types:

- `int` : Integer variable.

- `str` : String variable (default).

- `bool` : Bool variable.

`Returns` section contains the exit status of the function.

`Output` section describe the string printed to stdout.

## Versioning ##

\* JuNest uses the following [semantic versioning](http://semver.org/)

### Public API ###

The public API refers to the following parts of JuNest system:

- JuNest script CLI

Any potential code change that cause backwards incompatible changes to the

public API requires the major version to be incremented.