# Contributing

## Code Style

\* Tab indent code.

\* Keep code within 80 columns.

\* Global variables must be UPPERCASE. Temporary variables should be lowercase.

\* Use the `function` keyword for functions.

\* Quote all String variables.

\* Use `(( ))` for arithmetic expressions and `[[ ]]` otherwise.

\* Use `$(...)` instead of back-ticks.

\* Use `${path##\*/}` instead of `$(basename $path)`.

\* Use `${path%/\*}` instead of `$(dirname $path)`.

\* Prefer single-line expressions where appropriate:

[[ -n "$foo" ]] && other command

if [[ "$foo" == "bar" ]]; then command

elif [[ "$foo" == "baz" ]]; then other\_command

fi

case "$foo" in

bar) command ;;

baz) other\_command ;;

esac

## Pull Request Guidelines

\* Additional features should go in separate files within `share/chruby/`.

\* All new code must have [shunit2] unit-tests.

\* If a feature is out of scope or does more than switches Rubies,

it should become its own project. Simply copy the generic [Makefile]

for shell script projects.

### What Will Not Be Accepted

\* Completion: tab completion is not necessary for fuzzy matching

(ex: `chruby 1.9`).

\* Gemset functionality: there is already [chgems] and others, which provides

gemset like behavior.

\* Sub-commands: this tends to promote bloat. It was first popularized by [RVM]

and then copied by [rbenv].

\* [Bash][bash] or [Zsh][zsh] only features: chruby must fully support both

[bash] and [zsh].

\* New `.files`: we should strive to prevent dotfile proliferation, and instead

support common dotfiles which other Ruby Switchers could support.

\* New environment variables: much like with dotfile proliferation,

we should keep environment variable pollution to a minimum.

\* Ruby/OS specific workarounds: we should not tolerate the existence of bugs

in specific Rubies or Operating Systems. Instead we should strive to resolve

these bugs, to the benefit of all users.

\* Any [RVM]/[rbenv] specific feature: we should not blindly copy the features of

[RVM] or [rbenv] in the hopes of making new users comfortable. Instead,

we should question these features and rethink (or unthink) them.

\* Are some of these features even necessary?

\* Could they be implemented differently?

\* Do they over-optimize rarely performed tasks?

\* Do they make assumptions about workflow?

\* Do they add additional complexity?

\* Do they introduce possible Security risks?

[Makefile]: https://gist.github.com/3224049

[shunit2]: http://code.google.com/p/shunit2/

[bash]: http://www.gnu.org/software/bash/

[zsh]: http://www.zsh.org/

[RVM]: https://rvm.io/

[rbenv]: https://github.com/sstephenson/rbenv#readme

[chgems]: https://github.com/postmodern/chgems#readme