A lot of effort has been put into `faker.js` to create a useful and handy

library. There are still a lot of things to be done, so all contributions are

welcome! If you can make `faker.js` better, please read the following contribution guide.

# Important

\* Please make sure that you run both `gulp` and tests before making a PR.

## Support

`faker.js` relies on [commonJS](http://www.commonjs.org/) standard and supports both node.js and the

browsers. Keep this in mind, when modifying and/or extending the sources.

## Automation

\* The project is being built by [gulp](http://gulpjs.com/) (see [gulpfile](build/gulpfile.js)), destination directory is [build/build](build/build)

\* The documentation is auto-generated, based on [build/src](build/src) markdown sources. If you modify the main [Readme.md](Readme.md) file, the Pull Request will be rejected, since it will be overwritten by the upcoming `gulp` execution

\* The tests are executing `mocha` against all js contents of [test](test) directory

## Architecture

The sources are located in the [lib](lib) directory. All fake data generators are

divided into namespaces (each namespace being a separate module). Most of the

generators use the \*definitions\*, which are just plain JavaScript

objects/arrays/strings that are separate for each [locale](lib/locales).