Since gmaps.js has been created to accomplish some goals about simplicity and cleanliness, the contributions must be guided by some rules, in order to keep gmaps.js simple to use and brief in code.

## About requirements

gmaps.js only requires JavaScript, so new features must be written in plain JavaScript and shouldn’t depend of external libraries like jQuery, Prototype, etc.

## About examples and documentation

Big changes, like support for new libraries or services, or custom features, need an example file (or files, depending of the magnitude of the contribution) and an entry in the documentation page.

The example file must be included in the branch `gh-pages`, and be linked in the examples page. Example file in the `master` branch is optional.

The documentation must explain in an simple way any argument passed in the method, and tell if it’s argument is optional or not. No examples are required here.

## About new features

The main goal of gmaps.js is to have full support to all features of native Google Maps API, so the primary goal of the contributions to cover the missing features. But other contributions are welcome.

If the new feature has a functionality different from strictly work with maps (like animations, custom infoWindows or support for external services), it would be ideal to create an extension with this feature.

## About coding standards

\* Default values must be defined.

\* Functions must end with semi-colon.

\* Function names in camel case.

\* Two space for indentation.

\* Use strict mode syntax.

\* Don’t use prototype object to extend gmaps.js (except with extensions).

\* Use apply for calling functions.

\_\*\*Thanks for contributing with gmaps.js\*\*\_