## Contributing

I'm very open to contributions, big and small! For general instructions on submitting a pull request on GitHub, see these guides: [Fork A Repo](https://help.github.com/articles/fork-a-repo), and [Creating a pull request from a fork](https://help.github.com/articles/creating-a-pull-request-from-a-fork/).

### Changes to the plugin

In order to build the project, you will first need to install [npm](https://www.npmjs.org), and then globally install a package called evenizer with `npm install -g evenizer`. Then run `npm install` to install the project's dependencies, and then you should be good to run `npm run build` to build the project. At this point, the included `demo.html` should be working. You should make your changes in the `src` directory and be sure to run `npm run build` again before committing.

### Updating to a new version of libphonenumber

#### Step 1: Setup

Create a new dir (e.g. ~/workspace/libphonenumber-tools) where you will clone the libphonenumber project and a few other dependencies, and cd into it, and then:

```

git clone https://github.com/googlei18n/libphonenumber

git clone https://github.com/google/closure-library

git clone https://github.com/google/closure-compiler

git clone https://github.com/google/closure-linter

git clone https://github.com/google/python-gflags

```

Build Closure's compiler.jar in closure-compiler directory: `mvn -DskipTests` (requires maven to be installed - on MacOS, you can do this with `brew install maven`)

#### Step 2: Updating libphonenumber

Simply cd into the libphonenumber dir and checkout the required version tag e.g.

```

cd ~/workspace/libphonenumber-tools/libphonenumber

git checkout v8.9.14

```

Then to build the new version of utils.js:

1. Copy intl-tel-input/src/js/utils.js to libphonenumber/javascript/i18n/phonenumbers/demo.js

2. `ant -f libphonenumber/javascript/build.xml compile-demo` (requires ant to be installed - on MacOS, you can do this with `brew install ant`)

3. Copy libphonenumber/javascript/i18n/phonenumbers/demo-compiled.js to intl-tel-input/build/js/utils.js

Then, back in the intl-tel-input dir, first run the tests to make sure nothing has broken: `npm test`, then just commit the new utils.js, and create a pull request on Github.

### Updating the flag images

We get our flags from the region-flags project, which in turn pulls them in from Wikipedia. So cd into intl-tel-input/node\_modules/region-flags and then do the following:

1. Install some depenencies. On MacOS use brew: `brew install wget dos2unix librsvg optipng`

2. Run the make-aliases command: `./make-aliases.sh` (Note: I got some "No such file or directory" warnings)

3. Run the download command: `./download-wp.sh` (Note: this kept freezing for me, so I had to keep doing ctrl+c and then re-running it)

Finally, the last time I did this (October 2018) there was a problem with the Cayman Islands flag (region-flags/png/KY.png) - it should be aprx 1200x600px, like the other flags, but was instead tiny (36x36px), so I replaced it with [the old one from the region-flags repo](https://github.com/behdad/region-flags/blob/gh-pages/png/KY.png).

At this point, you should be good to cd back to the project root directory, and re-build the images with `npm run build:img`, and then check everything looks ok, and create a pull request on Github.

If when building, you get an error in the "exec:evenizer" task, you may need to temporarily increase the ulimit by running this command: `ulimit -S -n 2048`