Contributing to Marked

- [ ] Fork `markedjs/marked`.
- [ ] Clone the library locally using GitHub Desktop or the command line.
- [ ] Make sure you are on the `master` branch.
- [ ] Be sure to run `npm install` or `npm update`.
- [ ] Create a branch.
- [ ] Update code in `src` folder. (`lib` folder is for auto compiled code)
- [ ] Run `npm run test:all`, fix any broken things (for linting, you can run `npm run lint` to have the linter fix them for you).
- [ ] Run `npm run build:reset` to remove changes to compiled files.
- [ ] Submit a Pull Request.

Design principles

Marked tends to favor following the SOLID set of software design and development principles; mainly the single responsibility and open/closed principles:

- **Single responsibility:** Marked, and the components of Marked, have the single responsibility of converting Markdown strings into HTML.
- **Open/closed:** Marked favors giving developers the means to easily extend the library and its components over changing Marked's behavior through configuration options.

Priorities

We think we have our priorities sorted to build quality in.

The following table lists the ticket type labels we use when there is work to be done on the code either through an Issue or a PR; in priority order.

Ticket type label	Description
L0 - security	A security vulnerability within the Marked library is discovered.
L1 - broken	Valid usage results in incorrect output compared to supported specifications OR causes marked to crash AND there is no known workaround for the issue.
L2 - annoying	Similar to L1 - broken only there is a known workaround available for the issue.
RR - refactor and re-engineer	Results in an improvement to developers using Marked (improved readability) or end-users (faster performance) or both.
NFS - new feature (spec related)	A capability Marked does not currently provide but is in one of the supported specifications
NFU - new feature (user requested)	A capability Marked does not currently provide but has been requested by users of Marked.
NFE - new feature (should be an extension)	A capability Marked does not currently provide and is not part of a spec.

Test early, often, and everything

We try to write test cases to validate output (writing tests based on the supported specifications) and minimize regression (writing tests for issues fixed). Therefore, if you would like to contribute, some things you should know regarding the test harness.

Location	Description
/test/specs/commonmark	Tests for CommonMark compliance
/test/specs/gfm	Tests for GFM compliance
/test/specs/new	Tests not related to the original `markdown.pl`.
/test/specs/original	Tests validating against the original `markdown.pl`.
/test/specs/redos	Tests for ReDOS vulnerabilities

If your test uses features or options, assuming `gfm` is set to `false`, for example, you can add front-matter to the top of your `.md` file

```
---
gfm: false
---
```

Submitting PRs and Issues

Marked provides templates for submitting both pull requests and issues. When you begin creating a new PR or issue, you will see instructions on using the template.

The PR templates include checklists for both the submitter and the reviewer, which, in most cases, will not be the same person.

Scripts

When it comes to NPM commands, we try to use the native scripts provided by the NPM framework.

To run the tests:

```
npm test
```

To test whether you are using the standard syntax rules for the project:

```
npm run test:lint
```

To see time comparisons between Marked and other popular Markdown libraries:

```
npm run bench
```

To see the compiled rules from `src/rules.js`:

```
npm run rules
```

You can specify one or more `rule path`s to only show certain rules:

```
npm run rules -- block.gfm.item inline.pedantic.br

{
  block: {
```

```
  block: {
    gfm: {
      item: /^( *)((?:[*+-]|\\d{1,9}\\.)) ?[^\\n]*(?:\\n(?!\\1(?:[*+-]|\\d{1,9}\\.) ?)[^\\n]*)*/gm
    }
  },
  inline: {
    pedantic: {
      br: /^( {2,}|\\\\)\\n(?!\\s*$)/
    }
  }
}
```

To check for (and fix) standardized syntax (lint):

```
npm run lint
```

To build your own es5, esm, and minified versions of Marked:

```
npm run build
```