Contribution Guide for Manhattan Project

========================================

## Git Commit Guidelines

These guidelines have been copied from the [AngularJS](https://github.com/angular/angular.js/blob/master/CONTRIBUTING.md#-git-commit-guidelines)

project.

We have very precise rules over how our git commit messages can be formatted. This leads to \*\*more

readable messages\*\* that are easy to follow when looking through the \*\*project history\*\*. But also,

we use the git commit messages to \*\*generate the change log\*\*.

### Commit Message Format

Each commit message consists of a \*\*header\*\*, a \*\*body\*\* and a \*\*footer\*\*. The header has a special

format that includes a \*\*type\*\*, a \*\*scope\*\* and a \*\*subject\*\*:

```

<type>(<scope>): <subject>

<BLANK LINE>

<body>

<BLANK LINE>

<footer>

```

Any line of the commit message cannot be longer 100 characters! This allows the message to be easier

to read on github as well as in various git tools.

### Type

Must be one of the following:

\* \*\*feat\*\*: A new feature

\* \*\*fix\*\*: A bug fix

\* \*\*docs\*\*: Documentation only changes

\* \*\*style\*\*: Changes that do not affect the meaning of the code (white-space, formatting, missing

semi-colons, etc)

\* \*\*refactor\*\*: A code change that neither fixes a bug or adds a feature

\* \*\*perf\*\*: A code change that improves performance

\* \*\*test\*\*: Adding missing tests

\* \*\*chore\*\*: Changes to the build process or auxiliary tools and libraries such as documentation

generation

### Scope

In our project scope refers to BEM block which is touched by changes.

### Subject

The subject contains succinct description of the change:

\* use the imperative, present tense: "change" not "changed" nor "changes"

\* don't capitalize first letter

\* no dot (.) at the end

###Body

Just as in the \*\*subject\*\*, use the imperative, present tense: "change" not "changed" nor "changes"

The body should include the motivation for the change and contrast this with previous behavior.

###Footer

The footer should contain any information about \*\*Breaking Changes\*\* and is also the place to

reference GitHub issues that this commit \*\*Closes\*\*.

The last line of commits introducing breaking changes should be in the form `BREAKING\_CHANGE: <desc>`

A detailed explanation can be found in this [document][commit-message-format].

[commit-message-format]: https://docs.google.com/document/d/1QrDFcIiPjSLDn3EL15IJygNPiHORgU1\_OOAqWjiDU5Y/edit#

## Pull Request reviews comments

Following abbreviations should be used in PR merge comments:

\* `r+ {sha}`: review passed for all commits up to `{sha}`

\* `r-`: review revealed some errors, changes must be done to proceed

\* `r? @spanferov`: request for review from another user

\* `re-r? @spanferov`: request another user to do review again

\* `cc @spanferov, @smbd`: current message is also intended for...