# Contributing to atoum

The key words "\*\*must\*\*", "\*\*must not\*\*", "\*\*should\*\*", "\*\*should not\*\*" and "\*\*may\*\*" in this document are to be interpreted as described in [RFC 2119](http://www.ietf.org/rfc/rfc2119.txt).

## Opening Issues

Whenever you find a bug in atoum or think of a new feature you \*\*may\*\* open an issue in the bug tracker.

### Reporting bug

When reporting a bug, you \*\*should\*\* provide as much relevant informations as possible. This includes (but is not limited to):

\* PHP Version (`php -v`),

\* xDebug version (if loaded),

\* atoum version (sha1, phar, tag): get it with `bin/atoum --version` or `php atoum.phar --version` or `git rev-parse --short HEAD`,

\* A minimum (non) working example,

\* A clear description of the problem,

\* Expected and actual results.

You \*\*must\*\* also provide a descriptive title.

Here is an example issue:

```md

# Chaining multiple hasConstant assertions fails

| | |

|---------------|--------------|

| atoum version | `dev-master` |

| atoum SHA1 | `cbf9821` |

| PHP version | `PHP 5.3.29` |

| xDebug | not loaded |

When chaining multiple `hasConstant` assertions atoum reports the following error: `Asserter 'hasConstant' does not exist`

## Example

<?php

namespace atoum\shou\tests\units;

use atoum;

class runner extends atoum

{

public function testClass()

{

$this

->testedClass

->hasConstant('runStart')->isEqualTo('runnerStart')

->hasConstant('runStop')->isEqualTo('runnerStop')

->extends('atoum\shou\observable')

->implements('atoum\shou\observer')

;

}

}

## Actual result

There is 1 error:

=> atoum\shou\tests\units\runner::testClass():

==> Error E\_USER\_ERROR in code.php on line 13, generated by file test.php on line 37:

Asserter 'hasConstant' does not exist

## Expected result

Test should either pass or fail nicely, without a non-handled error.

```

### Requesting a new feature

When requesting a new feature you \*\*should\*\* ensute that it does not already exist in atoum or any extension.

You \*\*must\*\* provide a valid use case for the feature and it \*\*must\*\* be compliant with atoum's mantra (modern, simple and intuitive).

You \*\*should\*\* also provide code snippets to illustrate your need and how the feature will be used from a user point of view.

You \*\*may\*\* also link to external resources or other tools providing similar features to help reviewers.

## Sending Pull Requests

To be accepted your code contribution \*\*must\*\* follow atoum's [coding style](#coding-style).

You \*\*must\*\* write unit tests to cover your modifications.

Your commit history \*\*should\*\* be as clean as possible:

\* You \*\*should\*\* squash your commits to hide non-valudable commits (coding style fix, unit tests, …),

\* You \*\*should\*\* rebase your branch on top of master to make your pull-request fast-forward,

\* You \*\*must\*\* add a changelog entry.

You \*\*must\*\* also provide a descriptive title and a description for your changes. It \*\*may\*\* include code snippets to illustrate how to use

the feature you added.

## Coding style

Coding style respects [the PSR-2 recommendation](http://www.php-fig.org/psr/psr-2/), in addition to some other fixers provided by [the PHP CS Fixer tool](https://github.com/FriendsOfPhp/PHP-CS-Fixer).

To check the coding style, please run the following script:

```sh

$ vendor/bin/php-cs-fixer --dry-run --diff --verbose

```

### Compatibility

Your code \*\*must\*\* work on PHP from version 5.6 to the latest stable.

When you need to use a feature that is only available on PHP version greater than 5.6, you \*\*must\*\* either:

\* Use version sniffing (`version\_compare`),

\* Check if class exists,

\* Check if method exists,

\* …

You \*\*must\*\* ensure a consistent behavior on all supported version. If it is not possible, you \*\*must\*\* ensure a correct behavior.

### Naming Conventions

Function, variable, constant, class, interface and method \*\*must\*\* be named using `lowerCamelCase`.

Protected or private properties and methods \*\*must not\*\* use an underscore prefix.

### Global Variables

Global variables \*\*must not\*\* be used.

## Code of Conduct

By contributing to atoum you agree to adopt the [Open CoC](http://todogroup.org/opencodeofconduct/#atoum/team@atoum.org):

\* Be friendly and patient,

\* Be welcoming,

\* Be considerate,

\* Be respectful,

\* Be careful in the words that we choose,

\* Try to understand why we disagree.