# Changelog

All notable changes to this project will be documented in this file. This project adheres to [Semantic Versioning](https://semver.org/).

## x.y.z

### Added

### Changed

### Deprecated

### Removed

### Fixed

## 8.9.0: New features, bug fixes and deprecations

### Added

* RuleSet::removeMatchingRules() method (for the implementing classes AtRuleSet and DeclarationBlock) (#1249)
* RuleSet::removeAllRules() method (for the implementing classes AtRuleSet and DeclarationBlock) (#1249)
* Add Interface CSSElement (#1231)
* Methods getLineNumber and getColumnNumber which return a nullable int for the following classes: Comment, CSSList, SourceException, Charset, CSSNamespace, Import, Rule, DeclarationBlock, RuleSet, CSSFunction, Value (#1225, #1263)
* Positionable interface for CSS items that may have a position (line and perhaps column number) in the parsed CSS (#1221)

### Changed

* Parameters for getAllValues() are deconflated, so it now takes three (all optional), allowing $element and $ruleSearchPattern to be specified separately (#1241)
* Implement Positionable in the following CSS item classes: Comment, CSSList, SourceException, Charset, CSSNamespace, Import, Rule, DeclarationBlock, RuleSet, CSSFunction, Value (#1225)

### Deprecated

* Support for PHP < 7.2 is deprecated; version 9.0 will require PHP 7.2 or later (#1264)
* Passing a string or null to RuleSet::removeRule() is deprecated (implementing classes are AtRuleSet and DeclarationBlock); use removeMatchingRules() or removeAllRules() instead (#1249)
* Passing a Rule to RuleSet::getRules() or getRulesAssoc() is deprecated, affecting the implementing classes AtRuleSet and DeclarationBlock (call e.g. getRules($rule->getRule()) instead) (#1248)
* Passing a string as the first argument to getAllValues() is deprecated; the search pattern should now be passed as the second argument (#1241)
* Passing a Boolean as the second argument to getAllValues() is deprecated; the flag for searching in function arguments should now be passed as the third argument (#1241)
* getLineNo() is deprecated in these classes (use getLineNumber() instead): Comment, CSSList, SourceException, Charset, CSSNamespace, Import, Rule, DeclarationBlock, RuleSet, CSSFunction, Value (#1225, #1233)
* Rule::getColNo() is deprecated (use getColumnNumber() instead) (#1225, #1233)
* Providing zero as the line number argument to Rule::setPosition() is deprecated (pass null instead if there is no line number) (#1225, #1233)

### Fixed

* Set line number when RuleSet::addRule() called with only column number set (#1265)
* Ensure first rule added with RuleSet::addRule() has valid position (#1262)

## 8.8.0: Bug fixes and deprecations

### Added

* OutputFormat properties for space around specific list separators (#880)

### Changed

* Mark the OutputFormat constructor as @internal (#1131)
* Mark OutputFormatter as @internal (#896)
* Mark Selector::isValid() as @internal (#1037)
* Mark parsing-related methods of most CSS elements as @internal (#908)
* Mark OutputFormat::nextLevel() as @internal (#901)
* Make all non-private properties @internal (#886)

### Deprecated

* Deprecate extending OutputFormat (#1131)
* Deprecate OutputFormat::get() and ::set() (#1107)
* Deprecate support for -webkit-calc and -moz-calc (#1086)
* Deprecate magic method forwarding from OutputFormat to OutputFormatter (#894)
* Deprecate \_\_toString() (#1006)
* Deprecate greedy calculation of selector specificity (#1018)
* Deprecate the IE hack in Rule (#993, #1003)
* OutputFormat properties for space around list separators as an array (#880)
* Deprecate OutputFormat::level() (#870)

### Fixed

* Include comments for all rules in declaration block (#1169)
* Render rules in line and column number order (#1059)
* Create Size with correct types in expandBackgroundShorthand (#814)
* Parse @font-face src property as comma-delimited list (#794)

## 8.7.0: Add support for PHP 8.4

### Added

* Add support for PHP 8.4 (#675, #701, #746, #751)

### Changed

* Mark parsing-internal classes and methods as @internal (#711)
* Block installations on unsupported higher PHP versions (#691)

### Deprecated

* Deprecate the expansion of shorthand properties (#719)
* Deprecate Parser::setCharset() and Parser::getCharset() (#703)

### Fixed

* Fix type errors in PHP strict mode (#695)

## 8.6.0

### Added

* Support arithmetic operators in CSS function arguments (#607)
* Add support for inserting an item in a CSS list (#545)
* Add support for the dvh, lvh and svh length units (#415)

### Changed

* Improve performance of Value::parseValue with many delimiters by refactoring to remove array\_search() (#413)

## 8.5.2

### Changed

* Mark all class constants as @internal (#500)

### Fixed

* Fix undefined local variable in CalcFunction::parse() (#593)

## 8.5.1

### Fixed

* Fix PHP notice caused by parsing invalid color values having less than 6 characters (#485)
* Fix (regression) failure to parse at-rules with strict parsing (#456)

## 8.5.0

### Added

* Add a method to get an import's media queries (#384)
* Add more unit tests (#381, #382)

### Fixed

* Retain CSSList and Rule comments when rendering CSS (#351)
* Replace invalid turns unit with turn (#350)
* Also allow string values for rules (#348)
* Fix invalid calc parsing (#169)
* Handle scientific notation when parsing sizes (#179)
* Fix PHP 8.1 compatibility in ParserState::strsplit() (#344)

## 8.4.0

### Features

* Support for PHP 8.x
* PHPDoc annotations
* Allow usage of CSS variables inside color functions (by parsing them as regular functions)
* Use PSR-12 code style
* *No deprecations*

### Bugfixes

* Improved handling of whitespace in calc()
* Fix parsing units whose prefix is also a valid unit, like vmin
* Allow passing an object to CSSList#replace
* Fix PHP 7.3 warnings
* Correctly parse keyframes with %
* Don’t convert large numbers to scientific notation
* Allow a file to end after an @import
* Preserve case of CSS variables as specced
* Allow identifiers to use escapes the same way as strings
* No longer use eval for the comparison in getSelectorsBySpecificity, in case it gets passed untrusted input (CVE-2020-13756). Also fixed in 8.3.1, 8.2.1, 8.1.1, 8.0.1, 7.0.4, 6.0.2, 5.2.1, 5.1.3, 5.0.9, 4.0.1, 3.0.1, 2.0.1, 1.0.1.
* Prevent an infinite loop when parsing invalid grid line names
* Remove invalid unit vm
* Retain rule order after expanding shorthands

### Backwards-incompatible changes

* PHP ≥ 5.6 is now required
* HHVM compatibility target dropped

## 8.3.0 (2019-02-22)

* Refactor parsing logic to mostly reside in the class files whose data structure is to be parsed (this should eventually allow us to unit-test specific parts of the parsing logic individually).
* Fix error in parsing calc expessions when the first operand is a negative number, thanks to @raxbg.
* Support parsing CSS4 colors in hex notation with alpha values, thanks to @raxbg.
* Swallow more errors in lenient mode, thanks to @raxbg.
* Allow specifying arbitrary strings to output before and after declaration blocks, thanks to @westonruter.
* *No backwards-incompatible changes*
* *No deprecations*

## 8.2.0 (2018-07-13)

* Support parsing calc(), thanks to @raxbg.
* Support parsing grid-lines, again thanks to @raxbg.
* Support parsing legacy IE filters (progid:) in lenient mode, thanks to @FMCorz
* Performance improvements parsing large files, again thanks to @FMCorz
* *No backwards-incompatible changes*
* *No deprecations*

## 8.1.0 (2016-07-19)

* Comments are no longer silently ignored but stored with the object with which they appear (no render support, though). Thanks to @FMCorz.
* The IE hacks using \0 and \9 can now be parsed (and rendered) in lenient mode. Thanks (again) to @FMCorz.
* Media queries with or without spaces before the query are parsed. Still no *real* parsing support, though. Sorry…
* PHPUnit is now listed as a dev-dependency in composer.json.
* *No backwards-incompatible changes*
* *No deprecations*

## 8.0.0 (2016-06-30)

* Store source CSS line numbers in tokens and parsing exceptions.
* *No deprecations*

### Backwards-incompatible changes

* Unrecoverable parser errors throw an exception of type Sabberworm\CSS\Parsing\SourceException instead of \Exception.

## 7.0.3 (2016-04-27)

* Fixed parsing empty CSS when multibyte is off
* *No backwards-incompatible changes*
* *No deprecations*

## 7.0.2 (2016-02-11)

* 150 time performance boost thanks to @[ossinkine](https://github.com/ossinkine)
* *No backwards-incompatible changes*
* *No deprecations*

## 7.0.1 (2015-12-25)

* No more suppressed E\_NOTICE
* *No backwards-incompatible changes*
* *No deprecations*

## 7.0.0 (2015-08-24)

* Compatibility with PHP 7. Well timed, eh?
* *No deprecations*

### Backwards-incompatible changes

* The Sabberworm\CSS\Value\String class has been renamed to Sabberworm\CSS\Value\CSSString.

## 6.0.1 (2015-08-24)

* Remove some declarations in interfaces incompatible with PHP 5.3 (< 5.3.9)
* *No deprecations*

## 6.0.0 (2014-07-03)

* Format output using Sabberworm\CSS\OutputFormat
* *No backwards-incompatible changes*

### Deprecations

* The parse() method replaces \_\_toString with an optional argument (instance of the OutputFormat class)

## 5.2.0 (2014-06-30)

* Support removing a selector from a declaration block using $oBlock->removeSelector($mSelector)
* Introduce a specialized exception (Sabberworm\CSS\Parsing\OuputException) for exceptions during output rendering
* *No deprecations*

#### Backwards-incompatible changes

* Outputting a declaration block that has no selectors throws an OuputException instead of outputting an invalid {…} into the CSS document.

## 5.1.2 (2013-10-30)

* Remove the use of consumeUntil in comment parsing. This makes it possible to parse comments such as /\*\* Perfectly valid \*\*/
* Add fr relative size unit
* Fix some issues with HHVM
* *No backwards-incompatible changes*
* *No deprecations*

## 5.1.1 (2013-10-28)

* Updated CHANGELOG.md to reflect changes since 5.0.4
* *No backwards-incompatible changes*
* *No deprecations*

## 5.1.0 (2013-10-24)

* Performance enhancements by Michael M Slusarz
* More rescue entry points for lenient parsing (unexpected tokens between declaration blocks and unclosed comments)
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.8 (2013-08-15)

* Make default settings’ multibyte parsing option dependent on whether or not the mbstring extension is actually installed.
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.7 (2013-08-04)

* Fix broken decimal point output optimization
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.6 (2013-05-31)

* Fix broken unit test
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.5 (2013-04-17)

* Initial support for lenient parsing (setting this parser option will catch some exceptions internally and recover the parser’s state as neatly as possible).
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.4 (2013-03-21)

* Don’t output floats with locale-aware separator chars
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.3 (2013-03-21)

* More size units recognized
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.2 (2013-03-21)

* CHANGELOG.md file added to distribution
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.1 (2013-03-20)

* Internal cleanup
* *No backwards-incompatible changes*
* *No deprecations*

## 5.0.0 (2013-03-20)

* Correctly parse all known CSS 3 units (including Hz and kHz).
* Output RGB colors in short (#aaa or #ababab) notation
* Be case-insensitive when parsing identifiers.
* *No deprecations*

### Backwards-incompatible changes

* Sabberworm\CSS\Value\Color’s \_\_toString method overrides CSSList’s to maybe return something other than type(value, …) (see above).

## 4.0.0 (2013-03-19)

* Support for more @-rules
* Generic interface Sabberworm\CSS\Property\AtRule, implemented by all @-rule classes
* *No deprecations*

### Backwards-incompatible changes

* Sabberworm\CSS\RuleSet\AtRule renamed to Sabberworm\CSS\RuleSet\AtRuleSet
* Sabberworm\CSS\CSSList\MediaQuery renamed to Sabberworm\CSS\RuleSet\CSSList\AtRuleBlockList with differing semantics and API (which also works for other block-list-based @-rules like @supports).

## 3.0.0 (2013-03-06)

* Support for lenient parsing (on by default)
* *No deprecations*

### Backwards-incompatible changes

* All properties (like whether or not to use mb\_-functions, which default charset to use and – new – whether or not to be forgiving when parsing) are now encapsulated in an instance of Sabberworm\CSS\Settings which can be passed as the second argument to Sabberworm\CSS\Parser->\_\_construct().
* Specifying a charset as the second argument to Sabberworm\CSS\Parser->\_\_construct() is no longer supported. Use Sabberworm\CSS\Settings::create()->withDefaultCharset('some-charset') instead.
* Setting Sabberworm\CSS\Parser->bUseMbFunctions has no effect. Use Sabberworm\CSS\Settings::create()->withMultibyteSupport(true/false) instead.
* Sabberworm\CSS\Parser->parse() may throw a Sabberworm\CSS\Parsing\UnexpectedTokenException when in strict parsing mode.

## 2.0.0 (2013-01-29)

* Allow multiple rules of the same type per rule set

### Backwards-incompatible changes

* Sabberworm\CSS\RuleSet->getRules() returns an index-based array instead of an associative array. Use Sabberworm\CSS\RuleSet->getRulesAssoc() (which eliminates duplicate rules and lets the later rule of the same name win).
* Sabberworm\CSS\RuleSet->removeRule() works as it did before except when passed an instance of Sabberworm\CSS\Rule\Rule, in which case it would only remove the exact rule given instead of all the rules of the same type. To get the old behaviour, use Sabberworm\CSS\RuleSet->removeRule($oRule->getRule();

## 1.0

Initial release of a stable public API.

## 0.9

Last version not to use PSR-0 project organization semantics.