

8.15.0 (2025-06-08)

New features

Support using and `await` using syntax.

The AnyNode type is now defined in such a way that plugins can extend it.

Bug fixes

Fix an issue where the `bignum` property of literal nodes for non-decimal bigints had the wrong format.

The `acorn` CLI tool no longer crashes when emitting a tree that contains a bigint.

8.14.1 (2025-03-05)

Bug fixes

Fix an issue where await expressions in class field initializers were inappropriate allowed.
Properly allow await inside an async arrow function inside a class field initializer.
Mention the source file name in syntax error messages when
Property add an empty attributes property to every form
of ExportNamedDeclaration.

8.14.0 (2024-10-27)

New features

Support ES2025 import attributes.
Support ES2025 RegExp modifiers.

2.5.0 (2015-10-27)

Fixes

Fix tokenizer support in the command-line tool.

Stop allowing `new.target` outside of functions.

Remove legacy `guard` and `guardedHandler` properties from try nodes.

Stop allowing multiple `__proto__` properties on an object literal in strict mode.

Don't allow rest parameters to be non-identifier patterns.

Check for duplicate parameter names in arrow functions.

Bug fixes

Support some missing Unicode properties.

8.13.0 (2024-10-16)

New features

Upgrade to Unicode 16.0.

8.12.1 (2024-07-03)

Bug fixes

Fix a regression that caused Acorn to no longer run on Node versions <8.10.

2.6.0 (2015-11-09)	Fixes	New features
Don't crash when no options object is passed.	Support ES2025 duplicate capture group names in regular expressions.	Support Line numbers for escaped newlines in strings.
Don't crash when no options object is passed.	Support ES2025 duplicate capture group names in regular expressions.	Properly track line numbers for escaped newlines in strings.
Properly track line numbers for escaped newlines in strings.	Properly track line endings in raw strings of invalid template tokens.	Properly normalize line endings in raw strings of invalid template tokens.

New features

When `ecmaVersion` is set to 7, Acorn will parse the exponentiation operator (`**`).

The identifier character ranges are now based on Unicode 8.0.0.

Plugins can now override the `raiseRecoverable` method to override the way non-critical errors are handled.

2.7.0 (2016-01-04)

Fixes

Stop allowing rest parameters in setters.

Disallow `y` regexp flag in ES5.

Disallow `\00` and `\000` escapes in strict mode.

Raise an error when an import name is a reserved word.

8.11.3 (2023-12-29)

Bug fixes

Add `Function` and `Class` to the `AggregateType` type, so that they can be used in walkers without raising a type error.

Make sure `onToken` get an `import` keyword token when parsing `import.meta`.

Fix a bug where `.loc.start` could be undefined for `new.target` meta nodes.

8.11.2 (2023-10-27)

Bug fixes

Fix a bug that caused regular expressions after colon tokens to not be properly tokenized in some circumstances.

3.0.0 (2016-02-10)

Fixes

Fix bug that accidentally made undefined a reserved word when parsing ES7.

When parsing ES7.

3.0.2 (2016-02-10)

Fixes

Fix bug that accidentally made Undeleted head a reserved word
in parsing E7.

When parsing ES7.

3.0.0 (2016-02-10)

Breaking changes

The default value of the `ecmaVersion` option is now 6 (used to be 5). The support for comprehension syntax (which was dropped from the draft spec) has been removed.

Fixes

Let and yield are now “contextual keywords”, meaning you can mostly use them as identifiers in ES5 non-strict code. A parenthesized class or function expression after export default is now parsed correctly.

New features

Use a set of new, much more precise, TypeScript types.
Upgrade to Unicode 5.1.

Bug Fixes

- Fix an issue where a slash after a call to a property named the same as some keywords would be tokenized as a regular expression.
- Fix an issue where the tokenizer puts the tokens into an invalid state.
- Fix an issue where tokenizing (without parsing) an object literal with a property named class or function could, in some circumstance, put the tokenizer into an invalid state.

8.11.0 (2023-10-26)

Big fixes

Fix a regression where `oneloket` would receive `name` tokens for `'new'` keyword tokens.

New features

Visitors created with `visit.make` now have their base as *prototype*, rather than copying properties into a fresh object.

3.1.0 (2016-04-18)

Bug fixes

Properly tokenize the division operator directly after a function expression.

Allow trailing comma in destructuring arrays.

3.0.4 (2016-02-25)

Fixes

Allow update expressions as left-hand-side of the ES7 exponential operator.

8.10.0 (2023-07-05)

New features

Add a `checkPrivateFields` option that disables strict checking of private property use.

8.9.0 (2023-06-16)

Bug fixes

Forbid dynamic import after `new`, even when part of a member expression.

New features

Add Unicode properties for ES2023.

Add support for the `v` flag to regular expressions.

3.2.0 (2016-06-07)

New features

Fix bug in tokenizing of regexp operator after a function declaration.
 Fix a bug that caused `allowHashbang` to be set to false when not provided, even with `ecmaVersion >= 14`.
 Fix an exception when passing no option object to `parse` or `new Parser`.

Bug fixes

Improve handling of lack of unicode regexp support in host environment.
 Properly reject shorthand properties whose name is a keyword.

Bug fixes

Make type for `Comment` compatible with es6 types.

Bug fixes

8.8.1 (2022-10-24)

Fix incorrect parse error on `if (0) let\n[astral identifier char]`.
 Fix an exception when passing no option object to `parse` or `new Parser`.
 Fix `incorrectParseError` when `ecmaVersion <= 14`.
 Fix `incorrectParseError` on `if (0) let\n[astral identifier char]`.

3.3.0 (2016-07-25)

Bug fixes

Improve handling of lack of unicode regexp support in host environment.

Bug fixes

Properly reject shorthand properties whose name is a keyword.

Bug fixes

8.8.2 (2023-01-23)

A number of internal method signatures changed, so plugins might need to be updated.

8.8.0 (2022-07-21)

Bug fixes

The parser now raises errors on duplicated export names.

`arguments` and `eval` can now be used in shorthand properties.

Duplicate parameter names in non-simple argument lists now always produce an error.

New features

The `ecmaVersion` option now also accepts year-style version numbers (2015, etc).

Support for `async/await` syntax when `ecmaVersion` is ≥ 8 .

Support for trailing commas in call expressions when `ecmaVersion` is ≥ 8 .

Bug fixes

Allow parentheses around spread args in destructuring object assignment.

Fix an issue where the tree contained `directive` properties in when parsing with a language version that doesn't support them.

New features

Support hashbang comments by default in ECMAScript 2023 and later.

8.7.1 (2021-04-26)

Bug fixes

Stop handling "use strict" directives in ECMAScript versions before 5.

The default ECMAScript option value is now 7.

Breaking changes

4.0.0 (2016-08-07)

Fix false positives in duplicated export name errors.

Bug fixes

4.0.1 (2016-08-08)

Don't ignore period or `e` characters after octal numbers.
Fix broken parsing for call expressions in default parameter values of arrow functions.

Bug fixes

4.0.2 (2016-08-11)

Fix an issue where duplicate quoted export names in `export *` syntax were incorrectly checked.
Add missing type for `TokenType`s.

8.7.0 (2021-12-27)

Support quoted export names.
Upgrade to Unicode 14.
Add support for Unicode 13 properties in regular expressions.

New features

Use a loop to find line breaks, because the existing regexp search would overrun the end of the searched range and waste a lot of time in minified code.

Bug fixes

Disallow labeled declarations.

8.6.0 (2021-11-18)

4.0.4 (2016-12-19)

Bug fixes

Fix crash when `export` was followed by a keyword that can't be exported.

4.0.3 (2016-08-16)

Bug fixes

Allow regular function declarations inside single-statement `if` branches in loose mode. Forbid them entirely in strict mode.

Properly parse properties named `async` in ES2017 mode.

Fix bug where reserved words were broken in ES2017 mode.

Bug fixes

Fix a bug where an object literal with multiple `__proto__` properties would incorrectly be accepted if a later property value held an assignment.

New features

Support class private fields with the `in` operator.

8.5.0 (2021-09-06)

Bug fixes

Improve context-dependent tokenization in a number of corner cases.

Fix location tracking after a `0x2028` or `0x2029` character in a string literal (which before did not increase the line number).

Fix an issue where arrow function bodies in for loop context would inappropriately consume `in` operators.

Properly raise an error when a keyword contains a character escape.	Allow "use strict" to appear after other string literal expressions.
Don't allow the <code>asynch</code> keyword to be parenthesized.	Properly raise an error when a keyword contains a character escape.
Allow keywords as export names.	Allow "use strict" to appear after other string literal expressions.
Disallow parenthesized pattern expressions.	Properly raise an error when a keyword contains a character escape.

4.0.5 (2017-02-02)

Bug Fixes

Fix exponential behavior (manifesting itself as a complete hang for even relatively small source files) introduced by the new language feature <code>await</code> .	Fix a bug where <code>allowAwaitOutsideFunction</code> would allow <code>await</code> in class field initializers, and setting <code>ecmaVersion</code> to 13 or higher would allow top-level <code>await</code> in non-module sources.
---	--

4.0.6 (2017-02-02)

Bug Fixes

Don't raise an error when a function inside strict code has a non-simple parameter list.	Implement restriction that <code>for/of</code> loop LHS can't start with let.
---	--

Allow keywords as export names.	Fix a bug where <code>allowAwaitOutsideFunction</code> would allow <code>await</code> in class field initializers, and setting <code>ecmaVersion</code> to 13 or higher would allow top-level <code>await</code> in non-module sources.
Disallow parenthesized pattern expressions.	Allow "use strict" to appear after other string literal expressions.
Allow keywords as export names.	Allow "use strict" to appear after other string literal expressions.
Don't allow the <code>asynch</code> keyword to be parenthesized.	Properly raise an error when a keyword contains a character escape.
Properly raise an error when a keyword contains a character escape.	Properly raise an error when a keyword contains a character escape.

8.4.1 (2021-06-24)

Bug Fixes

Add support for ES2022 class static blocks.	Allow multiple input files to be passed to the CLI tool.
---	--

New Features

Fix wrong end locations stored on SequenceExpression nodes.	Implement restriction that <code>for/of</code> loop LHS can't start with let.
---	--

4.0.9 (2017-02-06)

Bug fixes

Fix incorrect error raised for parenthesized simple assignment targets, so that `(x) = 1` parses again.

4.0.8 (2017-02-03)

Bug fixes

Solve spurious parenthesized pattern errors by temporarily erring on the side of accepting programs that our delayed errors don't handle correctly yet.

4.0.7 (2017-02-02)

Bug fixes

Accept invalidly rejected code like `(x).y = 2` again.

8.4.0 (2021-06-11)

New features

A new option, `allowSuperOutsideMethod`, can be used to suppress the error when `super` is used in the wrong context.

8.3.0 (2021-05-31)

New features

Default `allowAwaitOutsideFunction` to `true` for ECMAScript 2022 and higher.

Add support for the `d` ([indices](#)) regexp flag.

8.2.4 (2021-05-04)

Bug fixes

Fix spec conformity in corner case ‘`for await (async of ...)`’.

8.2.3 (2021-05-04)

Bug fixes

Fix an issue where the library couldn't parse `for (async of ...)`.
 Fix a bug in UTF-16 decoding that would read characters incorrectly in some circumstances.

8.2.2 (2021-04-29)

Bug fixes

Fix a bug where a class field initialized to an async arrow function wouldn't allow await inside it. Same issue existed for generator arrow functions with `yield`.

4.0.10 (2017-02-07)

Bug fixes

Don't expect semicolons after default-exported functions or classes, even when they are expressions.
 Check for use of `'use strict'` directives in non-simple parameter functions, even when already in strict mode.

4.0.11 (2017-02-07)

Breaking changes

Parse declarations following `export default` as declaration nodes, not expressions. This means that class and function declarations nodes can now have null as their `id`.

Bug fixes

Allow all forms of member expressions to be parenthesized as value.

4.0.10 (2017-02-07)

Bug fixes

Don't expect semicolons after default-exported functions or classes, even when they are expressions.
 Check for use of `'use strict'` directives in non-simple parameter functions, even when already in strict mode.

5.0.2 (2017-03-30)

Bug fixes

A binary operator after a parenthesized arrow expression is no longer incorrectly treated as an error.

5.0.0 (2017-03-28)

Bug fixes

Raise an error for duplicated lexical bindings.

Fix spurious error when an assignment expression occurred after a spread expression.

Accept regular expressions after `of` (in `for/of`), `yield` (in a generator), and braced arrow functions.

Allow labels in front of `var` declarations, even in strict mode.

8.2.1 (2021-04-24)

Bug fixes

Fix a regression introduced in 8.2.0 where static or async class methods with keyword names fail to parse.

8.2.0 (2021-04-24)

New features

Add support for ES2022 class fields and private methods.

8.1.1 (2021-04-12)

Various

Stop shipping source maps in the NPM package.

8.1.0 (2021-03-09)

New features

Fix a spurious error in nested destructuring arrays.

Bug fixes

Expose `allowAnonymous` function in CLI interface.
Make `allowImportExportAnywhere` also apply to
`import.meta`.

5.0.3 (2017-04-01)

New features

Implement the [template literal revision proposal](#) for ES9.

Bug fixes

Fix spurious duplicate variable definition errors for named functions.

Adjust package.json to work with Node 12.16.0 and 13.0-13.6.

Bug fixes

8.0.5 (2021-01-25)

New features

Fix a spurious error in nested destructuring arrays.

Bug fixes

5.1.2 (2017-09-04)

Bug fixes

Disable parsing of legacy HTML-style comments in modules.
Fix parsing of async methods whose names are keywords.

8.0.4 (2020-10-05)

Bug fixes

Make `await x ** y` an error, following the spec.
Fix potentially exponential regular expression.

5.1.1 (2017-07-06)

Bug fixes

Fix problem with disambiguating regexp and division after a class.

8.0.3 (2020-10-02)

Bug fixes

Fix a wasteful loop during `Parser` creation when setting `ecmaVersion` to "latest".

5.1.0 (2017-07-05)

Bug fixes

Fix tokenizing of regexps in an object-desctructuring `for/of` loop and after `yield`.

8.0.2 (2020-09-30)

Bug fixes

Make the TypeScript types reflect the current allowed values for `ecmaVersion`.

Update to the set of Unicode properties from ES2021.
module in node 13+.

The package can now be loaded directly as an ECMAScript

New features

Disallow super when not in call or object position.
mode.

Make non-octal escape sequences a syntax error in strict

Disallow expressions like `(a = b) = c.`

Bug fixes

8.0.0 (2020-08-12)

Provide the correct value in the version export.

Bug fixes

8.0.1 (2020-08-12)

Fix another regexp/division tokenizer issue.

5.2.0 (2017-10-30)

Bug fixes

Fix a token context corruption bug.

Bug fixes

5.2.1 (2017-10-30)

Support `directive property` on directive expression
statements.

New features

Disallow super when not in call or object position.
mode.

Make non-octal escape sequences a syntax error in strict

Disallow expressions like `(a = b) = c.`

Allow shorthand properties `get` and `set` to be followed by
default values.
Make sure `%*` isn't parsed as a valid operator.
property-name position.

Fix token context tracking for class and function in

Bug fixes

Disallow super when not in call or object position.
mode.

Make non-octal escape sequences a syntax error in strict

Disallow expressions like `(a = b) = c.`

Allow shorthand properties `get` and `set` to be followed by
default values.
Make sure `%*` isn't parsed as a valid operator.
property-name position.

Fix token context tracking for class and function in

Bug fixes

Disallow super when not in call or object position.
mode.

Make non-octal escape sequences a syntax error in strict

Disallow expressions like `(a = b) = c.`

Allow shorthand properties `get` and `set` to be followed by
default values.
Make sure `%*` isn't parsed as a valid operator.
property-name position.

Fix token context tracking for class and function in

5.3.0 (2017-12-28)

Bug fixes

Fix parsing of floating point literals with leading zeroes in loose mode.

- Allow duplicate property names in object patterns.
- Don't allow static class methods named `prototype`.
- Disallow async functions directly under `if` or `else`.
- Parse right-hand-side of `for/of` as an assignment expression.
- Stricter parsing of `for/in`.
- Don't allow unicode escapes in contextual keywords.

New features

Parsing class members was factored into smaller methods to allow plugins to hook into it.

Breaking changes

The `ecmaVersion` option is now required. For the moment, omitting it will still work with a warning, but that will change in a future release.

Some changes to method signatures that may be used by plugins.

7.4.0 (2020-08-03)

New features

- Add support for logical assignment operators.
- Add support for numeric separators.

7.3.1 (2020-06-11)

Bug fixes

Make the string in the `version` export match the actual library version.

<h2>5.4.0 (2018-02-01)</h2> <p>New features</p> <ul style="list-style-type: none"> The identifier character categorization is now based on Unicode version 10. Acorn will now validate the content of regular expressions, including new ES9 features. <p>Bug fixes</p> <ul style="list-style-type: none"> Fix a bug that caused parsing of object patterns with a property named <code>set</code> that had a default value to fail. Add support for optional chaining (<code>? .</code>). 	<h2>7.2.0 (2020-05-09)</h2> <p>New features</p> <ul style="list-style-type: none"> Fix precedence issue in parsing of async arrow functions. Add support for nullish coalescing. Add support for import.meta. <p>Bug fixes</p> <ul style="list-style-type: none"> Fix a bug that caused parsing of object spread and rest. Add support for asynchronous iteration. Disallow octal escapes in strings in strict mode. Disallow duplicate or escaped flags on regular expressions.
<h2>5.5.0 (2018-02-27)</h2> <p>New features</p> <ul style="list-style-type: none"> Fix a bug that caused parsing of object patterns with a default value to fail. <p>Bug fixes</p> <ul style="list-style-type: none"> Add support for object spread and rest. Add support for array iteration. Add support for object iteration. Add support for object spread and rest. 	<h2>7.3.0 (2020-06-11)</h2> <p>New features</p> <ul style="list-style-type: none"> Fix a bug that caused parsing of object spread and rest. Add support for object iteration. Add support for object spread and rest. <p>Bug fixes</p> <ul style="list-style-type: none"> Fix a bug that caused parsing of object spread and rest. Add support for object iteration. Add support for object spread and rest.

5.5.3 (2018-03-08)

Bug fixes

A *second* republish of the code in 5.5.1, this time with yarn, to hopefully get valid timestamps.

5.5.2 (2018-03-08)

Bug fixes

A republish of the code in 5.5.1 in an attempt to solve an issue with the file timestamps in the npm package being 0.

5.5.1 (2018-03-06)

Bug fixes

Fix misleading error message for octal escapes in template strings.

Support `export * as ...` syntax.
Upgrade to Unicode 13.

6.4.1 (2020-03-09)

Bug fixes

More carefully check for valid UTF16 surrogate pairs in regexp validator.

7.1.1 (2020-03-01)

Bug fixes

Treat \8 and \9 as invalid escapes in template strings.
Allow unicode escapes in property names that are keywords.
Don't error on an exponential operator expression as argument to `await`.

More carefully check for valid UTF16 surrogate pairs in regexp validator.

5.7.0 (2018-06-15)

New features

Upgraded to Unicode 11.

5.6.0 (2018-05-31)

New features

10. Allow U+2028 and U+2029 in string when ECMAVersion >=
10. Allow binding-less catch statements when ECMAVersion >=
10. Add `allowAwaitOutsideFunction` option for parsing top-level await.

7.1.0 (2019-09-24)

Bug fixes

Disallow trailing object literal commas when ecmaversion is less than 5.

New features

Add a static `acorn` property to the `Parser` class that contains the entire module interface, to allow plugins to access the instance of the library that they are acting on.

7.0.0 (2019-08-13)

Breaking changes

Changes the node format for dynamic imports to use the `ImportExpression` node type, as defined in [ESTree](#). Makes 10 (ES2019) the default value for the `ecmaversion` option.

5.7.3 (2018-09-10)

Bug fixes

Fix failure to tokenize regexps after expressions like `x.of.`

Better error message for unterminated template literals.

5.7.2 (2018-08-24)

Bug fixes

Properly handle `allowAwaitOutsideFunction` in `for` statements.

Treat function declarations at the top level of modules like `let` bindings.

Don't allow `async` function declarations as the only statement under a label.

6.3.0 (2019-08-12)

New features

`sourceType: "module"` can now be used even when `ecmaVersion` is less than 6, to parse module-style code that otherwise conforms to an older standard.

6.2.1 (2019-07-21)

Bug fixes

Fix bug causing Acorn to treat some characters as identifier characters that shouldn't be treated as such.

Fix issue where setting the `allowReserved` option to `"never"` allowed reserved words in some circumstances.

6.0.0 (2018-09-14)

Bug fixes

- Better handle variable-redemption checks for catch bindings and functions directly under if statements.
- Forbidden new.target in top-level arrow functions.
- Fix issue with parsing a regexp after yield in some contexts.
- The package now comes with TypeScript definitions.

New features

- The default value of the ECMAScript option is now 9 (2018).
- Plugins work differently, and will have to be rewritten to work with this version.
- The loose parser and walker have been moved into separate packages (acorn-loose and acorn-walk).

Breaking changes

- The default value of the ECMAScript option is now 9 (2018).
- Plugins work differently, and will have to be rewritten to work with this version.
- The loose parser and walker have been moved into separate packages (acorn-loose and acorn-walk).

6.2.0 (2019-07-04)

Bug fixes

- Improve valid assignment checking in for/in and for/of loops.
- Disallow binding let in patterns.
- Support bigint syntax with ECMAScript >= 11.
- Support dynamic import syntax with ECMAScript >= 11.
- Upgrade to Unicode version 12.

New features

- Fix bug that caused parsing default exports of with names to fail.

6.1.1 (2019-02-27)

Bug fixes

- Fix bug that caused parsing default exports of with names to fail.

6.0.3 (2018-11-04)

Bug fixes

Fix bug in tokenizing an expression-less return followed by a function followed by a regular expression.

Remove stray symlink in the package tarball.

6.0.2 (2018-09-26)

Bug fixes

Fix bug where default expressions could fail to parse inside an object destructuring assignment expression.

6.0.1 (2018-09-14)

Bug fixes

Fix wrong value in `version` export.

6.1.0 (2019-02-08)

Bug fixes

Fix scope checking when redefining a `var` as a lexical binding.

New features

Split up `parseSubscripts` to use an internal `parseSubscript` method to make it easier to extend with plugins.

6.0.7 (2019-02-04)

Bug fixes

Check that exported bindings are defined.

Don't treat `\u180e` as a whitespace character.

Check for duplicate parameter names in methods.

Don't allow shorthand properties when they are generators or async methods.

6.0.5 (2019-01-02)

Bug Fixes

Fix TypeScript type for Parser.extend and add ALLOWAWAITOUTSIDEFUNCTION to options type.
Don't treat let as a keyword when the next token is { on the next line.
Fix bug that broke checking for parentheses around an object pattern in a destructuring assignment when preserveParens was on.

Further improvements to tokenizing regular expressions in corner cases.

6.0.4 (2018-11-05)

Bug Fixes

The content of class declarations and expressions is now always parsed in strict mode.

Don't allow let or const to bind the variable name let.
Treat class declarations as lexical.

Don't allow a generator function declaration as the sole body of an if or else.
Ignore "use strict" when after an empty statement.
Allow string line continuations with special line terminator characters.

Treat for bodies as part of the for scope when checking for conflicting bindings.
Fix bug with parsing yield in a for loop initializer.
Implement special cases around scope checking for functions.

6.0.6 (2019-01-30)

Bug Fixes

Forbidden binding await in async arrow function's parameter list.