postcss

7.0.18

[CLASSES](#gjdgxs)   [AtRule ▾](#30j0zll)

[#append](#1fob9te)   [#each](#3znysh7)   [#every](#2et92p0)   [#first](#tyjcwt)   [#index](#3dy6vkm)   [#insertAfter](#1t3h5sf)   [#insertBefore](#4d34og8)   [#last](#2s8eyo1)   [#prepend](#17dp8vu)   [#removeAll](#3rdcrjn)   [#removeChild](#26in1rg)   [#replaceValues](#lnxbz9)   [#some](#35nkun2)   [#walk](#1ksv4uv)   [#walkAtRules](#44sinio)   [#walkComments](#2jxsxqh)   [#walkDecls](#z337ya)   [#walkRules](#3j2qqm3)

[Comment ▾](#1y810tw)

[#after](#4i7ojhp)   [#before](#2xcytpi)   [#cleanRaws](#1ci93xb)   [#clone](#3whwml4)   [#cloneAfter](#2bn6wsx)   [#cloneBefore](#qsh70q)   [#error](#3as4poj)   [#next](#1pxezwc)   [#prev](#49x2ik5)   [#raw](#2p2csry)   [#remove](#147n2zr)   [#replaceWith](#3o7alnk)   [#root](#23ckvvd)   [#toString](#ihv636)   [#warn](#32hioqz)

[Container ▾](#1hmsyys)

[#after](#41mghml)   [#append](#2grqrue)   [#before](#vx1227)   [#cleanRaws](#3fwokq0)   [#clone](#1v1yuxt)   [#cloneAfter](#4f1mdlm)   [#cloneBefore](#2u6wntf)   [#each](#19c6y18)   [#error](#3tbugp1)   [#every](#28h4qwu)   [#first](#nmf14n)   [#index](#37m2jsg)   [#insertAfter](#1mrcu09)   [#insertBefore](#46r0co2)   [#last](#2lwamvv)   [#next](#111kx3o)   [#prepend](#3l18frh)   [#prev](#206ipza)   [#raw](#4k668n3)   [#remove](#2zbgiuw)   [#removeAll](#1egqt2p)   [#removeChild](#3ygebqi)   [#replaceValues](#2dlolyb)   [#replaceWith](#sqyw64)   [#root](#3cqmetx)   [#some](#1rvwp1q)   [#toString](#4bvk7pj)   [#walk](#2r0uhxc)   [#walkAtRules](#1664s55)   [#walkComments](#3q5sasy)   [#walkDecls](#25b2l0r)   [#walkRules](#kgcv8k)   [#warn](#34g0dwd)

[CssSyntaxError ▾](#1jlao46)

[#name](#43ky6rz)   [#reason](#2iq8gzs)   [#file](#xvir7l)   [#source](#3hv69ve)   [#plugin](#1x0gk37)   [#line](#4h042r0)   [#column](#2w5ecyt)   [#message](#1baon6m)   [#showSourceCode](#3vac5uf)   [#toString](#2afmg28)

[Declaration ▾](#pkwqa1)

[#after](#39kk8xu)   [#before](#1opuj5n)   [#cleanRaws](#48pi1tg)   [#clone](#2nusc19)   [#cloneAfter](#1302m92)   [#cloneBefore](#3mzq4wv)   [#error](#2250f4o)   [#next](#haapch)   [#prev](#319y80a)   [#raw](#1gf8i83)   [#remove](#40ew0vw)   [#replaceWith](#2fk6b3p)   [#root](#upglbi)   [#toString](#3ep43zb)   [#warn](#1tuee74)

[Input ▾](#4du1wux)

[#css](#2szc72q)   [#file](#184mhaj)   [#map](#3s49zyc)   [#id](#279ka65)   [#origin](#meukdy)   [#from](#36ei31r)

[LazyResult ▾](#1ljsd9k)

[#processor](#45jfvxd)   [#opts](#2koq656)   [#css](#zu0gcz)   [#content](#3jtnz0s)   [#map](#1yyy98l)   [#root](#4iylrwe)   [#messages](#2y3w247)   [#warnings](#1d96cc0)   [#toString](#3x8tuzt)   [#then](#2ce457m)   [#catch](#rjefff)   [#finally](#3bj1y38)

[Node ▾](#1qoc8b1)

[#error](#4anzqyu)   [#warn](#2pta16n)   [#remove](#14ykbeg)   [#toString](#3oy7u29)   [#clone](#243i4a2)   [#cloneBefore](#j8sehv)   [#cloneAfter](#338fx5o)   [#replaceWith](#1idq7dh)   [#next](#42ddq1a)   [#prev](#2hio093)   [#before](#wnyagw)   [#after](#3gnlt4p)   [#raw](#1vsw3ci)   [#root](#4fsjm0b)   [#cleanRaws](#2uxtw84)

[PreviousMap ▾](#1a346fx)

[#inline](#3u2rp3q)   [#consumer](#2981zbj)   [#withContent](#odc9jc)

[Processor ▾](#38czs75)

[#version](#1nia2ey)   [#plugins](#47hxl2r)   [#use](#2mn7vak)   [#process](#11si5id)

[Result ▾](#3ls5o66)

[#processor](#20xfydz)   [#messages](#4kx3h1s)   [#root](#302dr9l)   [#opts](#1f7o1he)   [#css](#3z7bk57)   [#map](#2eclud0)   [#toString](#thw4kt)   [#warn](#3dhjn8m)   [#warnings](#1smtxgf)   [#content](#4cmhg48)

[Root ▾](#2rrrqc1)

[#append](#16x20ju)   [#each](#3qwpj7n)   [#every](#261ztfg)   [#first](#l7a3n9)   [#index](#356xmb2)   [#insertAfter](#1kc7wiv)   [#insertBefore](#44bvf6o)   [#last](#2jh5peh)   [#on](#ymfzma)   [#prepend](#3im3ia3)   [#removeAll](#1xrdshw)   [#removeChild](#4hr1b5p)   [#replaceValues](#2wwbldi)   [#some](#1c1lvlb)   [#toResult](#3w19e94)   [#walk](#2b6jogx)   [#walkAtRules](#qbtyoq)   [#walkComments](#3abhhcj)   [#walkDecls](#1pgrrkc)   [#walkRules](#49gfa85)

[Rule ▾](#2olpkfy)

[#append](#13qzunr)   [#each](#3nqndbk)   [#every](#22vxnjd)   [#first](#i17xr6)   [#index](#320vgez)   [#insertAfter](#1h65qms)   [#insertBefore](#415t9al)   [#last](#2gb3jie)   [#prepend](#vgdtq7)   [#removeAll](#3fg1ce0)   [#removeChild](#1ulbmlt)   [#replaceValues](#4ekz59m)   [#selectors](#2tq9fhf)   [#some](#18vjpp8)   [#walk](#3sv78d1)   [#walkAtRules](#280hiku)   [#walkComments](#n5rssn)   [#walkDecls](#375fbgg)   [#walkRules](#1maplo9)

[Warning ▾](#46ad4c2)

[#type](#2lfnejv)   [#text](#10kxoro)   [#line](#3kkl7fh)   [#column](#1zpvhna)   [#toString](#4jpj0b3)

[NAMESPACES](#2yutaiw)   [list ▾](#1e03kqp)

[.space](#3xzr3ei)   [.comma](#2d51dmb)

[postcss ▾](#sabnu4)

[.plugin](#3c9z6hx)   [.stringify](#1rf9gpq)   [.parse](#4bewzdj)   [.vendor](#2qk79lc)   [.list](#15phjt5)   [.comment](#3pp52gy)   [.atRule](#24ufcor)   [.decl](#jzpmwk)   [.rule](#33zd5kd)   [.root](#1j4nfs6)

[vendor ▾](#434ayfz)

[.prefix](#2i9l8ns)   [.unprefixed](#xevivl)

[GLOBAL](#3hej1je)

[Need help reading this?](https://documentation.js.org/reading-documentation.html)

CLASSES

Extends [Container](#1hmsyys)  [lib/at-rule.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/at-rule.js#L21-L90)

### [**AtRule (defaults)**](#30j0zll)

Represents an at-rule.

If it’s followed in the CSS by a {} block, this node will have a nodes property representing its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | any |  |

#### Examples

const root = postcss.parse('@charset "UTF-8"; @media print {}')  
  
const charset = root.first  
charset.type //=> 'atrule'  
charset.nodes //=> undefined  
  
const media = root.last  
media.nodes //=> []

#### Instance Members

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L322-L331)

### [**append (children)**](#1fob9te)

Inserts new nodes to the end of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.append(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L63-L89)

### [**each (callback)**](#3znysh7)

Iterates through the container’s immediate children, calling callback for each child.

Returning false in the callback will break iteration.

This method only iterates through the container’s immediate children. If you need to recursively iterate through all the container’s descendant nodes, use [Container#walk](#2r0uhxc).

Unlike the for {}-cycle or Array#forEach this iterator is safe if you are mutating the array of child nodes during iteration. PostCSS will adjust the current index to match the mutations.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

const root = postcss.parse('a { color: black; z-index: 1 }')  
const rule = root.first  
  
for (const decl of rule.nodes) {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Cycle will be infinite, because cloneBefore moves the current node  
 // to the next index  
}  
  
rule.each(decl => {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Will be executed only for color and z-index  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L539-L541)

### [**every (condition)**](#2et92p0)

Returns true if callback returns true for all of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is every child pass condition.

Examples

const noPrefixes = rule.every(i => i.prop[0] !== '-')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L582-L585)

### [**first ()**](#tyjcwt)

The container’s first child.

Type: [Node](#1qoc8b1)

#### Examples

rule.first === rules.nodes[0]

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L568-L572)

### [**index (child)**](#3dy6vkm)

Returns a child’s index within the [Container#nodes](http://docs.google.com/Container#nodes) array.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | [Node](#1qoc8b1) | Child of the current container. |

#### Returns

#### [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) : Child index.

Examples

rule.index( rule.nodes[2] ) //=> 2

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L414-L431)

### [**insertAfter (exist, add)**](#1t3h5sf)

Insert new node after old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

[Node](#1qoc8b1) : This node for methods chain.

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L386-L404)

### [**insertBefore (exist, add)**](#4d34og8)

Insert new node before old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.insertBefore(decl, decl.clone({ prop: '-webkit-' + decl.prop }))

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L595-L598)

### [**last ()**](#2s8eyo1)

The container’s last child.

Type: [Node](#1qoc8b1)

#### Examples

rule.last === rule.nodes[rule.nodes.length - 1]

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L353-L366)

### [**prepend (children)**](#17dp8vu)

Inserts new nodes to the start of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.prepend(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L475-L482)

### [**removeAll ()**](#3rdcrjn)

Removes all children from the container and cleans their parent properties.

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.removeAll()  
rule.nodes.length //=> 0

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L447-L463)

### [**removeChild (child)**](#26in1rg)

Removes node from the container and cleans the parent properties from the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain

Examples

rule.nodes.length //=> 5  
rule.removeChild(decl)  
rule.nodes.length //=> 4  
decl.parent //=> undefined

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L510-L526)

### [**replaceValues (pattern, opts, callback)**](#lnxbz9)

Passes all declaration values within the container that match pattern through callback, replacing those values with the returned result of callback.

This method is useful if you are using a custom unit or function and need to iterate through all values.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| pattern | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp)) | Replace pattern. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | Options to speed up the search. |
| opts.props | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)>) | An array of property names. |
| opts.fast | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | String that’s used to narrow down values and speed up the regexp search. |
| callback | ([function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)) | String to replace pattern or callback that returns a new value. The callback will receive the same arguments as those passed to a function parameter of String#replace . |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

root.replaceValues(/\d+rem/, { fast: 'rem' }, string => {  
 return 15 \* parseInt(string) + 'px'  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L554-L556)

### [**some (condition)**](#35nkun2)

Returns true if callback returns true for (at least) one of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is some child pass condition.

Examples

const hasPrefix = rule.some(i => i.prop[0] === '-')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L110-L124)

### [**walk (callback)**](#1ksv4uv)

Traverses the container’s descendant nodes, calling callback for each node.

Like container.each(), this method is safe to use if you are mutating arrays during iteration.

If you only need to iterate through the container’s immediate children, use [Container#each](#19c6y18).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walk(node => {  
 // Traverses all descendant nodes.  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L255-L276)

### [**walkAtRules (name?, callback)**](#44sinio)

Traverses the container’s descendant nodes, calling callback for each at-rule node.

If you pass a filter, iteration will only happen over at-rules that have matching names.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| name | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter at-rules by name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkAtRules(rule => {  
 if (isOld(rule.name)) rule.remove()  
})  
  
let first = false  
root.walkAtRules('charset', rule => {  
 if (!first) {  
 first = true  
 } else {  
 rule.remove()  
 }  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L294-L300)

### [**walkComments (callback)**](#2jxsxqh)

Traverses the container’s descendant nodes, calling callback for each comment node.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkComments(comment => {  
 comment.remove()  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L155-L176)

### [**walkDecls (prop?, callback)**](#z337ya)

Traverses the container’s descendant nodes, calling callback for each declaration node.

If you pass a filter, iteration will only happen over declarations with matching properties.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter declarations by property name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkDecls(decl => {  
 checkPropertySupport(decl.prop)  
})  
  
root.walkDecls('border-radius', decl => {  
 decl.remove()  
})  
  
root.walkDecls(/^background/, decl => {  
 decl.value = takeFirstColorFromGradient(decl.value)  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L201-L223)

### [**walkRules (selector?, callback)**](#3j2qqm3)

Traverses the container’s descendant nodes, calling callback for each rule node.

If you pass a filter, iteration will only happen over rules with matching selectors.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| selector | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter rules by selector. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : returns false if iteration was broke.

Examples

const selectors = []  
root.walkRules(rule => {  
 selectors.push(rule.selector)  
})  
console.log(`Your CSS uses ${ selectors.length } selectors`)

Extends [Node](#1qoc8b1)  [lib/comment.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/comment.js#L11-L34)

### [**Comment (defaults)**](#1y810tw)

Represents a comment between declarations or statements (rule and at-rules).

Comments inside selectors, at-rule parameters, or declaration values will be stored in the raws properties explained above.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | any |  |

#### Instance Members

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L313-L316)

### [**after (add)**](#4i7ojhp)

Insert new node after current node to current node’s parent.

Just alias for node.parent.insertAfter(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.after('color: black')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L296-L299)

### [**before (add)**](#2xcytpi)

Insert new node before current node to current node’s parent.

Just alias for node.parent.insertBefore(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.before('content: ""')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L394-L398)

### [**cleanRaws (keepBetween?)**](#1ci93xb)

Clear the code style properties for the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| keepBetween | [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean)? | Keep the raws.between symbols. |

#### Returns

#### [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined) :

Examples

node.raws.before //=> ' '  
node.cleanRaws()  
node.raws.before //=> undefined

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L183-L189)

### [**clone (overrides = {})**](#3whwml4)

Returns an exact clone of the node.

The resulting cloned node and its (cloned) children will retain code style properties.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : Clone of the node.

Examples

decl.raws.before //=> "\n "  
const cloned = decl.clone({ prop: '-moz-' + decl.prop })  
cloned.raws.before //=> "\n "  
cloned.toString() //=> -moz-transform: scale(0)

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L216-L220)

### [**cloneAfter (overrides = {})**](#2bn6wsx)

Shortcut to clone the node and insert the resulting cloned node after the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

[Node](#1qoc8b1) : New node.

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L202-L206)

### [**cloneBefore (overrides = {})**](#qsh70q)

Shortcut to clone the node and insert the resulting cloned node before the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Mew properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : New node

Examples

decl.cloneBefore({ prop: '-moz-' + decl.prop })

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L89-L95)

### [**error (message, opts = {})**](#3as4poj)

Returns a CssSyntaxError instance containing the original position of the node in the source, showing line and column numbers and also a small excerpt to facilitate debugging.

If present, an input source map will be used to get the original position of the source, even from a previous compilation step (e.g., from Sass compilation).

This method produces very useful error messages.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| message | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Error description. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Options. |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this error. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the error. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the error. |

#### Returns

#### [CssSyntaxError](#1jlao46) : Error object to throw it.

Examples

if (!variables[name]) {  
 throw decl.error('Unknown variable ' + name, { word: name })  
 // CssSyntaxError: postcss-vars:a.sass:4:3: Unknown variable $black  
 // color: $black  
 // a  
 // ^  
 // background: white  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L260-L264)

### [**next ()**](#1pxezwc)

Returns the next child of the node’s parent. Returns undefined if the current node is the last child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Next node.

Examples

if (comment.text === 'delete next') {  
 const next = comment.next()  
 if (next) {  
 next.remove()  
 }  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L278-L282)

### [**prev ()**](#49x2ik5)

Returns the previous child of the node’s parent. Returns undefined if the current node is the first child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Previous node.

Examples

const annotation = decl.prev()  
if (annotation.type === 'comment') {  
 readAnnotation(annotation.text)  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L363-L366)

### [**raw (prop, defaultType?)**](#2p2csry)

Returns a [Node#raws](http://docs.google.com/Node#raws) value. If the node is missing the code style property (because the node was manually built or cloned), PostCSS will try to autodetect the code style property by looking at other nodes in the tree.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Name of code style property. |
| defaultType | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)? | Name of default value, it can be missed if the value is the same as prop. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Code style value.

Examples

const root = postcss.parse('a { background: white }')  
root.nodes[0].append({ prop: 'color', value: 'black' })  
root.nodes[0].nodes[1].raws.before //=> undefined  
root.nodes[0].nodes[1].raw('before') //=> ' '

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L139-L145)

### [**remove ()**](#147n2zr)

Removes the node from its parent and cleans the parent properties from the node and its children.

#### Returns

#### [Node](#1qoc8b1) : Node to make calls chain.

Examples

if (decl.prop.match(/^-webkit-/)) {  
 decl.remove()  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L234-L244)

### [**replaceWith (nodes)**](#3o7alnk)

Inserts node(s) before the current node and removes the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| nodes | ...[Node](#1qoc8b1) | Mode(s) to replace current one. |

#### Returns

#### [Node](#1qoc8b1) : Current node to methods chain.

Examples

if (atrule.name === 'mixin') {  
 atrule.replaceWith(mixinRules[atrule.params])  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L376-L380)

### [**root ()**](#23ckvvd)

Finds the Root instance of the node’s tree.

#### Returns

#### [Root](#2rrrqc1) : Root parent.

Examples

root.nodes[0].nodes[0].root() === root

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L158-L165)

### [**toString (stringifier = stringify)**](#ihv636)

Returns a CSS string representing the node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| stringifier | ([stringifier](#2vor4mt) | [syntax](#1au1eum))? = stringify | A syntax to use in string generation. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : CSS string of this node.

Examples

postcss.rule({ selector: 'a' }).toString() //=> "a {}"

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L122-L126)

### [**warn (result, text, opts?)**](#32hioqz)

This method is provided as a convenience wrapper for [Result#warn](#3dhjn8m).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| result | [Result](#3ls5o66) | The [Result](#3ls5o66) instance that will receive the warning. |
| text | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Warning message. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Options |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this warning. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the warning. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the warning. |

#### Returns

#### [Warning](#46ad4c2) : Created warning object.

Examples

const plugin = postcss.plugin('postcss-deprecated', () => {  
 return (root, result) => {  
 root.walkDecls('bad', decl => {  
 decl.warn(result, 'Deprecated property bad')  
 })  
 }  
})

Extends [Node](#1qoc8b1)  [lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L23-L693)

### [**Container ()**](#1hmsyys)

The [Root](#2rrrqc1), [AtRule](#30j0zll), and [Rule](#2olpkfy) container nodes inherit some common methods to help work with their children.

Note that all containers can store any content. If you write a rule inside a rule, PostCSS will parse it.

#### Instance Members

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L313-L316)

### [**after (add)**](#41mghml)

Insert new node after current node to current node’s parent.

Just alias for node.parent.insertAfter(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.after('color: black')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L322-L331)

### [**append (children)**](#2grqrue)

Inserts new nodes to the end of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.append(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L296-L299)

### [**before (add)**](#vx1227)

Insert new node before current node to current node’s parent.

Just alias for node.parent.insertBefore(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.before('content: ""')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L394-L398)

### [**cleanRaws (keepBetween?)**](#3fwokq0)

Clear the code style properties for the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| keepBetween | [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean)? | Keep the raws.between symbols. |

#### Returns

#### [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined) :

Examples

node.raws.before //=> ' '  
node.cleanRaws()  
node.raws.before //=> undefined

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L183-L189)

### [**clone (overrides = {})**](#1v1yuxt)

Returns an exact clone of the node.

The resulting cloned node and its (cloned) children will retain code style properties.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : Clone of the node.

Examples

decl.raws.before //=> "\n "  
const cloned = decl.clone({ prop: '-moz-' + decl.prop })  
cloned.raws.before //=> "\n "  
cloned.toString() //=> -moz-transform: scale(0)

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L216-L220)

### [**cloneAfter (overrides = {})**](#4f1mdlm)

Shortcut to clone the node and insert the resulting cloned node after the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

[Node](#1qoc8b1) : New node.

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L202-L206)

### [**cloneBefore (overrides = {})**](#2u6wntf)

Shortcut to clone the node and insert the resulting cloned node before the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Mew properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : New node

Examples

decl.cloneBefore({ prop: '-moz-' + decl.prop })

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L63-L89)

### [**each (callback)**](#19c6y18)

Iterates through the container’s immediate children, calling callback for each child.

Returning false in the callback will break iteration.

This method only iterates through the container’s immediate children. If you need to recursively iterate through all the container’s descendant nodes, use [Container#walk](#2r0uhxc).

Unlike the for {}-cycle or Array#forEach this iterator is safe if you are mutating the array of child nodes during iteration. PostCSS will adjust the current index to match the mutations.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

const root = postcss.parse('a { color: black; z-index: 1 }')  
const rule = root.first  
  
for (const decl of rule.nodes) {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Cycle will be infinite, because cloneBefore moves the current node  
 // to the next index  
}  
  
rule.each(decl => {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Will be executed only for color and z-index  
})

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L89-L95)

### [**error (message, opts = {})**](#3tbugp1)

Returns a CssSyntaxError instance containing the original position of the node in the source, showing line and column numbers and also a small excerpt to facilitate debugging.

If present, an input source map will be used to get the original position of the source, even from a previous compilation step (e.g., from Sass compilation).

This method produces very useful error messages.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| message | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Error description. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Options. |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this error. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the error. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the error. |

#### Returns

#### [CssSyntaxError](#1jlao46) : Error object to throw it.

Examples

if (!variables[name]) {  
 throw decl.error('Unknown variable ' + name, { word: name })  
 // CssSyntaxError: postcss-vars:a.sass:4:3: Unknown variable $black  
 // color: $black  
 // a  
 // ^  
 // background: white  
}

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L539-L541)

### [**every (condition)**](#28h4qwu)

Returns true if callback returns true for all of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is every child pass condition.

Examples

const noPrefixes = rule.every(i => i.prop[0] !== '-')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L582-L585)

### [**first ()**](#nmf14n)

The container’s first child.

Type: [Node](#1qoc8b1)

#### Examples

rule.first === rules.nodes[0]

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L568-L572)

### [**index (child)**](#37m2jsg)

Returns a child’s index within the [Container#nodes](http://docs.google.com/Container#nodes) array.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | [Node](#1qoc8b1) | Child of the current container. |

#### Returns

#### [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) : Child index.

Examples

rule.index( rule.nodes[2] ) //=> 2

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L414-L431)

### [**insertAfter (exist, add)**](#1mrcu09)

Insert new node after old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

[Node](#1qoc8b1) : This node for methods chain.

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L386-L404)

### [**insertBefore (exist, add)**](#46r0co2)

Insert new node before old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.insertBefore(decl, decl.clone({ prop: '-webkit-' + decl.prop }))

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L595-L598)

### [**last ()**](#2lwamvv)

The container’s last child.

Type: [Node](#1qoc8b1)

#### Examples

rule.last === rule.nodes[rule.nodes.length - 1]

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L260-L264)

### [**next ()**](#111kx3o)

Returns the next child of the node’s parent. Returns undefined if the current node is the last child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Next node.

Examples

if (comment.text === 'delete next') {  
 const next = comment.next()  
 if (next) {  
 next.remove()  
 }  
}

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L353-L366)

### [**prepend (children)**](#3l18frh)

Inserts new nodes to the start of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.prepend(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L278-L282)

### [**prev ()**](#206ipza)

Returns the previous child of the node’s parent. Returns undefined if the current node is the first child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Previous node.

Examples

const annotation = decl.prev()  
if (annotation.type === 'comment') {  
 readAnnotation(annotation.text)  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L363-L366)

### [**raw (prop, defaultType?)**](#4k668n3)

Returns a [Node#raws](http://docs.google.com/Node#raws) value. If the node is missing the code style property (because the node was manually built or cloned), PostCSS will try to autodetect the code style property by looking at other nodes in the tree.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Name of code style property. |
| defaultType | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)? | Name of default value, it can be missed if the value is the same as prop. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Code style value.

Examples

const root = postcss.parse('a { background: white }')  
root.nodes[0].append({ prop: 'color', value: 'black' })  
root.nodes[0].nodes[1].raws.before //=> undefined  
root.nodes[0].nodes[1].raw('before') //=> ' '

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L139-L145)

### [**remove ()**](#2zbgiuw)

Removes the node from its parent and cleans the parent properties from the node and its children.

#### Returns

#### [Node](#1qoc8b1) : Node to make calls chain.

Examples

if (decl.prop.match(/^-webkit-/)) {  
 decl.remove()  
}

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L475-L482)

### [**removeAll ()**](#1egqt2p)

Removes all children from the container and cleans their parent properties.

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.removeAll()  
rule.nodes.length //=> 0

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L447-L463)

### [**removeChild (child)**](#3ygebqi)

Removes node from the container and cleans the parent properties from the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain

Examples

rule.nodes.length //=> 5  
rule.removeChild(decl)  
rule.nodes.length //=> 4  
decl.parent //=> undefined

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L510-L526)

### [**replaceValues (pattern, opts, callback)**](#2dlolyb)

Passes all declaration values within the container that match pattern through callback, replacing those values with the returned result of callback.

This method is useful if you are using a custom unit or function and need to iterate through all values.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| pattern | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp)) | Replace pattern. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | Options to speed up the search. |
| opts.props | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)>) | An array of property names. |
| opts.fast | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | String that’s used to narrow down values and speed up the regexp search. |
| callback | ([function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)) | String to replace pattern or callback that returns a new value. The callback will receive the same arguments as those passed to a function parameter of String#replace . |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

root.replaceValues(/\d+rem/, { fast: 'rem' }, string => {  
 return 15 \* parseInt(string) + 'px'  
})

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L234-L244)

### [**replaceWith (nodes)**](#sqyw64)

Inserts node(s) before the current node and removes the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| nodes | ...[Node](#1qoc8b1) | Mode(s) to replace current one. |

#### Returns

#### [Node](#1qoc8b1) : Current node to methods chain.

Examples

if (atrule.name === 'mixin') {  
 atrule.replaceWith(mixinRules[atrule.params])  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L376-L380)

### [**root ()**](#3cqmetx)

Finds the Root instance of the node’s tree.

#### Returns

#### [Root](#2rrrqc1) : Root parent.

Examples

root.nodes[0].nodes[0].root() === root

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L554-L556)

### [**some (condition)**](#1rvwp1q)

Returns true if callback returns true for (at least) one of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is some child pass condition.

Examples

const hasPrefix = rule.some(i => i.prop[0] === '-')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L158-L165)

### [**toString (stringifier = stringify)**](#4bvk7pj)

Returns a CSS string representing the node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| stringifier | ([stringifier](#2vor4mt) | [syntax](#1au1eum))? = stringify | A syntax to use in string generation. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : CSS string of this node.

Examples

postcss.rule({ selector: 'a' }).toString() //=> "a {}"

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L110-L124)

### [**walk (callback)**](#2r0uhxc)

Traverses the container’s descendant nodes, calling callback for each node.

Like container.each(), this method is safe to use if you are mutating arrays during iteration.

If you only need to iterate through the container’s immediate children, use [Container#each](#19c6y18).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walk(node => {  
 // Traverses all descendant nodes.  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L255-L276)

### [**walkAtRules (name?, callback)**](#1664s55)

Traverses the container’s descendant nodes, calling callback for each at-rule node.

If you pass a filter, iteration will only happen over at-rules that have matching names.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| name | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter at-rules by name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkAtRules(rule => {  
 if (isOld(rule.name)) rule.remove()  
})  
  
let first = false  
root.walkAtRules('charset', rule => {  
 if (!first) {  
 first = true  
 } else {  
 rule.remove()  
 }  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L294-L300)

### [**walkComments (callback)**](#3q5sasy)

Traverses the container’s descendant nodes, calling callback for each comment node.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkComments(comment => {  
 comment.remove()  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L155-L176)

### [**walkDecls (prop?, callback)**](#25b2l0r)

Traverses the container’s descendant nodes, calling callback for each declaration node.

If you pass a filter, iteration will only happen over declarations with matching properties.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter declarations by property name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkDecls(decl => {  
 checkPropertySupport(decl.prop)  
})  
  
root.walkDecls('border-radius', decl => {  
 decl.remove()  
})  
  
root.walkDecls(/^background/, decl => {  
 decl.value = takeFirstColorFromGradient(decl.value)  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L201-L223)

### [**walkRules (selector?, callback)**](#kgcv8k)

Traverses the container’s descendant nodes, calling callback for each rule node.

If you pass a filter, iteration will only happen over rules with matching selectors.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| selector | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter rules by selector. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : returns false if iteration was broke.

Examples

const selectors = []  
root.walkRules(rule => {  
 selectors.push(rule.selector)  
})  
console.log(`Your CSS uses ${ selectors.length } selectors`)

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L122-L126)

### [**warn (result, text, opts?)**](#34g0dwd)

This method is provided as a convenience wrapper for [Result#warn](#3dhjn8m).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| result | [Result](#3ls5o66) | The [Result](#3ls5o66) instance that will receive the warning. |
| text | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Warning message. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Options |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this warning. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the warning. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the warning. |

#### Returns

#### [Warning](#46ad4c2) : Created warning object.

Examples

const plugin = postcss.plugin('postcss-deprecated', () => {  
 return (root, result) => {  
 root.walkDecls('bad', decl => {  
 decl.warn(result, 'Deprecated property bad')  
 })  
 }  
})

Extends [Error](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Error)  [lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L33-L243)

### [**CssSyntaxError (message, line?, column?, source?, file?, plugin?)**](#1jlao46)

The CSS parser throws this error for broken CSS.

Custom parsers can throw this error for broken custom syntax using the [Node#error](#4anzqyu) method.

PostCSS will use the input source map to detect the original error location. If you wrote a Sass file, compiled it to CSS and then parsed it with PostCSS, PostCSS will show the original position in the Sass file.

If you need the position in the PostCSS input (e.g., to debug the previous compiler), use error.input.file.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| message | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Error message. |
| line | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)? | Source line of the error. |
| column | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)? | Source column of the error. |
| source | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)? | Source code of the broken file. |
| file | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)? | Absolute path to the broken file. |
| plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)? | PostCSS plugin name, if error came from plugin. |

#### Examples

// Catching and checking syntax error  
try {  
 postcss.parse('a{')  
} catch (error) {  
 if (error.name === 'CssSyntaxError') {  
 error //=> CssSyntaxError  
 }  
}

// Raising error from plugin  
throw node.error('Unknown variable', { plugin: 'postcss-vars' })

#### Instance Members

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L58-L58)

### [**name ()**](#43ky6rz)

Always equal to 'CssSyntaxError'. You should always check error type by error.name === 'CssSyntaxError' instead of error instanceof CssSyntaxError, because npm could have several PostCSS versions.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

if (error.name === 'CssSyntaxError') {  
 error //=> CssSyntaxError  
}

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L67-L67)

### [**reason ()**](#2iq8gzs)

Error message.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

error.message //=> 'Unclosed block'

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L79-L79)

### [**file ()**](#xvir7l)

Absolute path to the broken file.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

error.file //=> 'a.sass'  
error.input.file //=> 'a.css'

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L91-L91)

### [**source ()**](#3hv69ve)

Source code of the broken file.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

error.source //=> 'a { b {} }'  
error.input.column //=> 'a b { }'

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L102-L102)

### [**plugin ()**](#1x0gk37)

Plugin name, if error came from plugin.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

error.plugin //=> 'postcss-vars'

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L114-L114)

### [**line ()**](#4h042r0)

Source line of the error.

Type: [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)

#### Examples

error.line //=> 2  
error.input.line //=> 4

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L124-L124)

### [**column ()**](#2w5ecyt)

Source column of the error.

Type: [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)

#### Examples

error.column //=> 1  
error.input.column //=> 4

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L144-L144)

### [**message ()**](#1baon6m)

Full error text in the GNU error format with plugin, file, line and column.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

error.message //=> 'a.css:1:1: Unclosed block'

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L173-L211)

### [**showSourceCode (color?)**](#3vac5uf)

Returns a few lines of CSS source that caused the error.

If the CSS has an input source map without sourceContent, this method will return an empty string.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| color | [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean)? | Whether arrow will be colored red by terminal color codes. By default, PostCSS will detect color support by process.stdout.isTTY and process.env.NODE\_DISABLE\_COLORS . |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Few lines of CSS source that caused the error.

Examples

error.showSourceCode() //=> " 4 | }  
 // 5 | a {  
 // > 6 | bad  
 // | ^  
 // 7 | }  
 // 8 | b {"

[lib/css-syntax-error.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/css-syntax-error.js#L223-L229)

### [**toString ()**](#2afmg28)

Returns error position, message and source code of the broken part.

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Error position, message and source code.

Examples

error.toString() //=> "CssSyntaxError: app.css:1:1: Unclosed block  
 // > 1 | a {  
 // | ^"

Extends [Node](#1qoc8b1)  [lib/declaration.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/declaration.js#L14-L75)

### [**Declaration (defaults)**](#pkwqa1)

Represents a CSS declaration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | any |  |

#### Examples

const root = postcss.parse('a { color: black }')  
const decl = root.first.first  
decl.type //=> 'decl'  
decl.toString() //=> ' color: black'

#### Instance Members

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L313-L316)

### [**after (add)**](#39kk8xu)

Insert new node after current node to current node’s parent.

Just alias for node.parent.insertAfter(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.after('color: black')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L296-L299)

### [**before (add)**](#1opuj5n)

Insert new node before current node to current node’s parent.

Just alias for node.parent.insertBefore(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.before('content: ""')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L394-L398)

### [**cleanRaws (keepBetween?)**](#48pi1tg)

Clear the code style properties for the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| keepBetween | [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean)? | Keep the raws.between symbols. |

#### Returns

#### [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined) :

Examples

node.raws.before //=> ' '  
node.cleanRaws()  
node.raws.before //=> undefined

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L183-L189)

### [**clone (overrides = {})**](#2nusc19)

Returns an exact clone of the node.

The resulting cloned node and its (cloned) children will retain code style properties.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : Clone of the node.

Examples

decl.raws.before //=> "\n "  
const cloned = decl.clone({ prop: '-moz-' + decl.prop })  
cloned.raws.before //=> "\n "  
cloned.toString() //=> -moz-transform: scale(0)

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L216-L220)

### [**cloneAfter (overrides = {})**](#1302m92)

Shortcut to clone the node and insert the resulting cloned node after the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

[Node](#1qoc8b1) : New node.

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L202-L206)

### [**cloneBefore (overrides = {})**](#3mzq4wv)

Shortcut to clone the node and insert the resulting cloned node before the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Mew properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : New node

Examples

decl.cloneBefore({ prop: '-moz-' + decl.prop })

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L89-L95)

### [**error (message, opts = {})**](#2250f4o)

Returns a CssSyntaxError instance containing the original position of the node in the source, showing line and column numbers and also a small excerpt to facilitate debugging.

If present, an input source map will be used to get the original position of the source, even from a previous compilation step (e.g., from Sass compilation).

This method produces very useful error messages.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| message | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Error description. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Options. |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this error. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the error. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the error. |

#### Returns

#### [CssSyntaxError](#1jlao46) : Error object to throw it.

Examples

if (!variables[name]) {  
 throw decl.error('Unknown variable ' + name, { word: name })  
 // CssSyntaxError: postcss-vars:a.sass:4:3: Unknown variable $black  
 // color: $black  
 // a  
 // ^  
 // background: white  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L260-L264)

### [**next ()**](#haapch)

Returns the next child of the node’s parent. Returns undefined if the current node is the last child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Next node.

Examples

if (comment.text === 'delete next') {  
 const next = comment.next()  
 if (next) {  
 next.remove()  
 }  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L278-L282)

### [**prev ()**](#319y80a)

Returns the previous child of the node’s parent. Returns undefined if the current node is the first child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Previous node.

Examples

const annotation = decl.prev()  
if (annotation.type === 'comment') {  
 readAnnotation(annotation.text)  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L363-L366)

### [**raw (prop, defaultType?)**](#1gf8i83)

Returns a [Node#raws](http://docs.google.com/Node#raws) value. If the node is missing the code style property (because the node was manually built or cloned), PostCSS will try to autodetect the code style property by looking at other nodes in the tree.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Name of code style property. |
| defaultType | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)? | Name of default value, it can be missed if the value is the same as prop. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Code style value.

Examples

const root = postcss.parse('a { background: white }')  
root.nodes[0].append({ prop: 'color', value: 'black' })  
root.nodes[0].nodes[1].raws.before //=> undefined  
root.nodes[0].nodes[1].raw('before') //=> ' '

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L139-L145)

### [**remove ()**](#40ew0vw)

Removes the node from its parent and cleans the parent properties from the node and its children.

#### Returns

#### [Node](#1qoc8b1) : Node to make calls chain.

Examples

if (decl.prop.match(/^-webkit-/)) {  
 decl.remove()  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L234-L244)

### [**replaceWith (nodes)**](#2fk6b3p)

Inserts node(s) before the current node and removes the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| nodes | ...[Node](#1qoc8b1) | Mode(s) to replace current one. |

#### Returns

#### [Node](#1qoc8b1) : Current node to methods chain.

Examples

if (atrule.name === 'mixin') {  
 atrule.replaceWith(mixinRules[atrule.params])  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L376-L380)

### [**root ()**](#upglbi)

Finds the Root instance of the node’s tree.

#### Returns

#### [Root](#2rrrqc1) : Root parent.

Examples

root.nodes[0].nodes[0].root() === root

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L158-L165)

### [**toString (stringifier = stringify)**](#3ep43zb)

Returns a CSS string representing the node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| stringifier | ([stringifier](#2vor4mt) | [syntax](#1au1eum))? = stringify | A syntax to use in string generation. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : CSS string of this node.

Examples

postcss.rule({ selector: 'a' }).toString() //=> "a {}"

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L122-L126)

### [**warn (result, text, opts?)**](#1tuee74)

This method is provided as a convenience wrapper for [Result#warn](#3dhjn8m).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| result | [Result](#3ls5o66) | The [Result](#3ls5o66) instance that will receive the warning. |
| text | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Warning message. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Options |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this warning. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the warning. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the warning. |

#### Returns

#### [Warning](#46ad4c2) : Created warning object.

Examples

const plugin = postcss.plugin('postcss-deprecated', () => {  
 return (root, result) => {  
 root.walkDecls('bad', decl => {  
 decl.warn(result, 'Deprecated property bad')  
 })  
 }  
})

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L14-L167)

### [**Input (css, opts = {})**](#4du1wux)

Represents the source CSS.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| css | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Input CSS source. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | [Processor#process](#11si5id) options. |

#### Examples

const root = postcss.parse(css, { from: file })  
const input = root.source.input

#### Instance Members

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L33-L33)

### [**css ()**](#2szc72q)

Input CSS source

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

const input = postcss.parse('a{}', { from: file }).input  
input.css //=> "a{}"

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L54-L54)

### [**file ()**](#184mhaj)

The absolute path to the CSS source file defined with the from option.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

const root = postcss.parse(css, { from: 'a.css' })  
root.source.input.file //=> '/home/ai/a.css'

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L71-L71)

### [**map ()**](#3s49zyc)

The input source map passed from a compilation step before PostCSS (for example, from Sass compiler).

Type: [PreviousMap](#1a346fx)

#### Examples

root.source.input.map.consumer().sources //=> ['a.sass']

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L88-L88)

### [**id ()**](#279ka65)

The unique ID of the CSS source. It will be created if from option is not provided (because PostCSS does not know the file path).

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

const root = postcss.parse(css)  
root.source.input.file //=> undefined  
root.source.input.id //=> "<input css 8LZeVF>"

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L125-L142)

### [**origin (line, column)**](#meukdy)

Reads the input source map and returns a symbol position in the input source (e.g., in a Sass file that was compiled to CSS before being passed to PostCSS).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| line | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | Line in input CSS. |
| column | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | Column in input CSS. |

#### Returns

#### [filePosition](#3utoxif) : Position in input source.

Examples

root.source.input.origin(1, 1) //=> { file: 'a.css', line: 3, column: 1 }

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L164-L166)

### [**from ()**](#36ei31r)

The CSS source identifier. Contains [Input#file](#184mhaj) if the user set the from option, or [Input#id](#279ka65) if they did not.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

const root = postcss.parse(css, { from: 'a.css' })  
root.source.input.from //=> "/home/ai/a.css"  
  
const root = postcss.parse(css)  
root.source.input.from //=> "<input css 1>"

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L44-L427)

### [**LazyResult (processor, css, opts)**](#1ljsd9k)

A Promise proxy for the result of PostCSS transformations.

A LazyResult instance is returned by [Processor#process](#11si5id).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| processor | any |  |
| css | any |  |
| opts | any |  |

#### Examples

const lazy = postcss([autoprefixer]).process(css)

#### Instance Members

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L81-L83)

### [**processor ()**](#45jfvxd)

Returns a [Processor](#38czs75) instance, which will be used for CSS transformations.

Type: [Processor](#38czs75)

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L90-L92)

### [**opts ()**](#2koq656)

Options from the [Processor#process](#11si5id) call.

Type: [processOptions](#29yz7q8)

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L106-L108)

### [**css ()**](#zu0gcz)

Processes input CSS through synchronous plugins, converts Root to a CSS string and returns [Result#css](#3z7bk57).

This property will only work with synchronous plugins. If the processor contains any asynchronous plugins it will throw an error. This is why this method is only for debug purpose, you should always use [LazyResult#then](#2ce457m).

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L122-L124)

### [**content ()**](#3jtnz0s)

An alias for the css property. Use it with syntaxes that generate non-CSS output.

This property will only work with synchronous plugins. If the processor contains any asynchronous plugins it will throw an error. This is why this method is only for debug purpose, you should always use [LazyResult#then](#2ce457m).

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L138-L140)

### [**map ()**](#1yyy98l)

Processes input CSS through synchronous plugins and returns [Result#map](#2eclud0).

This property will only work with synchronous plugins. If the processor contains any asynchronous plugins it will throw an error. This is why this method is only for debug purpose, you should always use [LazyResult#then](#2ce457m).

Type: SourceMapGenerator

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L155-L157)

### [**root ()**](#4iylrwe)

Processes input CSS through synchronous plugins and returns [Result#root](#302dr9l).

This property will only work with synchronous plugins. If the processor contains any asynchronous plugins it will throw an error.

This is why this method is only for debug purpose, you should always use [LazyResult#then](#2ce457m).

Type: [Root](#2rrrqc1)

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L172-L174)

### [**messages ()**](#2y3w247)

Processes input CSS through synchronous plugins and returns [Result#messages](#4kx3h1s).

This property will only work with synchronous plugins. If the processor contains any asynchronous plugins it will throw an error.

This is why this method is only for debug purpose, you should always use [LazyResult#then](#2ce457m).

Type: [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Message](#p49hy1)>

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L182-L184)

### [**warnings ()**](#1d96cc0)

Processes input CSS through synchronous plugins and calls [Result#warnings()](http://docs.google.com/Result#warnings()).

#### Returns

[Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Warning](#46ad4c2)> : Warnings from plugins.

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L194-L196)

### [**toString ()**](#3x8tuzt)

Alias for the [LazyResult#css](#zu0gcz) property.

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Output CSS.

Examples

lazy + '' === lazy.css

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L216-L227)

### [**then (onFulfilled, onRejected)**](#2ce457m)

Processes input CSS through synchronous and asynchronous plugins and calls onFulfilled with a Result instance. If a plugin throws an error, the onRejected callback will be executed.

It implements standard Promise API.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| onFulfilled | [onFulfilled](#393x0lu) | Callback will be executed when all plugins will finish work. |
| onRejected | [onRejected](#1o97atn) | Callback will be executed on any error. |

#### Returns

#### [Promise](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Promise) : Promise API to make queue.

Examples

postcss([autoprefixer]).process(css, { from: cssPath }).then(result => {  
 console.log(result.css)  
})

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L246-L248)

### [**catch (onRejected)**](#rjefff)

Processes input CSS through synchronous and asynchronous plugins and calls onRejected for each error thrown in any plugin.

It implements standard Promise API.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| onRejected | [onRejected](#1o97atn) | Callback will be executed on any error. |

#### Returns

#### [Promise](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Promise) : Promise API to make queue.

Examples

postcss([autoprefixer]).process(css).then(result => {  
 console.log(result.css)  
}).catch(error => {  
 console.error(error)  
})

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L266-L268)

### [**finally (onFinally)**](#3bj1y38)

Processes input CSS through synchronous and asynchronous plugins and calls onFinally on any error or when all plugins will finish work.

It implements standard Promise API.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| onFinally | onFinally | Callback will be executed on any error or when all plugins will finish work. |

#### Returns

#### [Promise](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Promise) : Promise API to make queue.

Examples

postcss([autoprefixer]).process(css).finally(() => {  
 console.log('processing ended')  
})

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L35-L564)

### [**Node (defaults = {})**](#1qoc8b1)

All node classes inherit the following common methods.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Value for node properties. |

#### Instance Members

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L89-L95)

### [**error (message, opts = {})**](#4anzqyu)

Returns a CssSyntaxError instance containing the original position of the node in the source, showing line and column numbers and also a small excerpt to facilitate debugging.

If present, an input source map will be used to get the original position of the source, even from a previous compilation step (e.g., from Sass compilation).

This method produces very useful error messages.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| message | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Error description. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Options. |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this error. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the error. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the error. |

#### Returns

#### [CssSyntaxError](#1jlao46) : Error object to throw it.

Examples

if (!variables[name]) {  
 throw decl.error('Unknown variable ' + name, { word: name })  
 // CssSyntaxError: postcss-vars:a.sass:4:3: Unknown variable $black  
 // color: $black  
 // a  
 // ^  
 // background: white  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L122-L126)

### [**warn (result, text, opts?)**](#2pta16n)

This method is provided as a convenience wrapper for [Result#warn](#3dhjn8m).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| result | [Result](#3ls5o66) | The [Result](#3ls5o66) instance that will receive the warning. |
| text | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Warning message. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Options |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Plugin name that created this warning. PostCSS will set it automatically. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | A word inside a node’s string that should be highlighted as the source of the warning. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | An index inside a node’s string that should be highlighted as the source of the warning. |

#### Returns

#### [Warning](#46ad4c2) : Created warning object.

Examples

const plugin = postcss.plugin('postcss-deprecated', () => {  
 return (root, result) => {  
 root.walkDecls('bad', decl => {  
 decl.warn(result, 'Deprecated property bad')  
 })  
 }  
})

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L139-L145)

### [**remove ()**](#14ykbeg)

Removes the node from its parent and cleans the parent properties from the node and its children.

#### Returns

#### [Node](#1qoc8b1) : Node to make calls chain.

Examples

if (decl.prop.match(/^-webkit-/)) {  
 decl.remove()  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L158-L165)

### [**toString (stringifier = stringify)**](#3oy7u29)

Returns a CSS string representing the node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| stringifier | ([stringifier](#2vor4mt) | [syntax](#1au1eum))? = stringify | A syntax to use in string generation. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : CSS string of this node.

Examples

postcss.rule({ selector: 'a' }).toString() //=> "a {}"

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L183-L189)

### [**clone (overrides = {})**](#243i4a2)

Returns an exact clone of the node.

The resulting cloned node and its (cloned) children will retain code style properties.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : Clone of the node.

Examples

decl.raws.before //=> "\n "  
const cloned = decl.clone({ prop: '-moz-' + decl.prop })  
cloned.raws.before //=> "\n "  
cloned.toString() //=> -moz-transform: scale(0)

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L202-L206)

### [**cloneBefore (overrides = {})**](#j8sehv)

Shortcut to clone the node and insert the resulting cloned node before the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Mew properties to override in the clone. |

#### Returns

#### [Node](#1qoc8b1) : New node

Examples

decl.cloneBefore({ prop: '-moz-' + decl.prop })

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L216-L220)

### [**cloneAfter (overrides = {})**](#338fx5o)

Shortcut to clone the node and insert the resulting cloned node after the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| overrides | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | New properties to override in the clone. |

#### Returns

[Node](#1qoc8b1) : New node.

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L234-L244)

### [**replaceWith (nodes)**](#1idq7dh)

Inserts node(s) before the current node and removes the current node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| nodes | ...[Node](#1qoc8b1) | Mode(s) to replace current one. |

#### Returns

#### [Node](#1qoc8b1) : Current node to methods chain.

Examples

if (atrule.name === 'mixin') {  
 atrule.replaceWith(mixinRules[atrule.params])  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L260-L264)

### [**next ()**](#42ddq1a)

Returns the next child of the node’s parent. Returns undefined if the current node is the last child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Next node.

Examples

if (comment.text === 'delete next') {  
 const next = comment.next()  
 if (next) {  
 next.remove()  
 }  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L278-L282)

### [**prev ()**](#2hio093)

Returns the previous child of the node’s parent. Returns undefined if the current node is the first child.

#### Returns

#### ([Node](#1qoc8b1) | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Previous node.

Examples

const annotation = decl.prev()  
if (annotation.type === 'comment') {  
 readAnnotation(annotation.text)  
}

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L296-L299)

### [**before (add)**](#wnyagw)

Insert new node before current node to current node’s parent.

Just alias for node.parent.insertBefore(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.before('content: ""')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L313-L316)

### [**after (add)**](#3gnlt4p)

Insert new node after current node to current node’s parent.

Just alias for node.parent.insertAfter(node, add).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

decl.after('color: black')

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L363-L366)

### [**raw (prop, defaultType?)**](#1vsw3ci)

Returns a [Node#raws](http://docs.google.com/Node#raws) value. If the node is missing the code style property (because the node was manually built or cloned), PostCSS will try to autodetect the code style property by looking at other nodes in the tree.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Name of code style property. |
| defaultType | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)? | Name of default value, it can be missed if the value is the same as prop. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Code style value.

Examples

const root = postcss.parse('a { background: white }')  
root.nodes[0].append({ prop: 'color', value: 'black' })  
root.nodes[0].nodes[1].raws.before //=> undefined  
root.nodes[0].nodes[1].raw('before') //=> ' '

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L376-L380)

### [**root ()**](#4fsjm0b)

Finds the Root instance of the node’s tree.

#### Returns

#### [Root](#2rrrqc1) : Root parent.

Examples

root.nodes[0].nodes[0].root() === root

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L394-L398)

### [**cleanRaws (keepBetween?)**](#2uxtw84)

Clear the code style properties for the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| keepBetween | [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean)? | Keep the raws.between symbols. |

#### Returns

#### [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined) :

Examples

node.raws.before //=> ' '  
node.cleanRaws()  
node.raws.before //=> undefined

[lib/previous-map.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/previous-map.js#L24-L140)

### [**PreviousMap (css, opts?)**](#1a346fx)

Source map information from input CSS. For example, source map after Sass compiler.

This class will automatically find source map in input CSS or in file system near input file (according from option).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| css | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Input CSS source. |
| opts | [processOptions](#29yz7q8)? | [Processor#process](#11si5id) options. |

#### Examples

const root = postcss.parse(css, { from: 'a.sass.css' })  
root.input.map //=> PreviousMap

#### Instance Members

[lib/previous-map.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/previous-map.js#L36-L36)

### [**inline ()**](#3u2rp3q)

Was source map inlined by data-uri to input CSS.

Type: [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean)

[lib/previous-map.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/previous-map.js#L52-L57)

### [**consumer ()**](#2981zbj)

Create a instance of SourceMapGenerator class from the source-map library to work with source map information.

It is lazy method, so it will create object only on first call and then it will use cache.

#### Returns

SourceMapGenerator : Object with source map information.

[lib/previous-map.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/previous-map.js#L64-L67)

### [**withContent ()**](#odc9jc)

Does source map contains sourcesContent with input source text.

#### Returns

[boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is sourcesContent present.

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L12-L135)

### [**Processor (plugins)**](#38czs75)

Contains plugins to process CSS. Create one Processor instance, initialize its plugins, and then use that instance on numerous CSS files.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| plugins | ([Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<([Plugin](#488uthg) | [pluginFunction](#2ne53p9))> | [Processor](#38czs75)) = [] | PostCSS plugins. See [Processor#use](#2mn7vak) for plugin format. |

#### Examples

const processor = postcss([autoprefixer, precss])  
processor.process(css1).then(result => console.log(result.css))  
processor.process(css2).then(result => console.log(result.css))

#### Instance Members

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L28-L28)

### [**version ()**](#1nia2ey)

Current PostCSS version.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

if (result.processor.version.split('.')[0] !== '6') {  
 throw new Error('This plugin works only with PostCSS 6')  
}

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L38-L38)

### [**plugins ()**](#47hxl2r)

Plugins added to this processor.

Type: [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[pluginFunction](#2ne53p9)>

#### Examples

const processor = postcss([autoprefixer, precss])  
processor.plugins.length //=> 2

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L70-L73)

### [**use (plugin)**](#2mn7vak)

Adds a plugin to be used as a CSS processor.

PostCSS plugin can be in 4 formats:

* A plugin created by [postcss.plugin](#3c9z6hx) method.
* A function. PostCSS will pass the function a @{link Root} as the first argument and current [Result](#3ls5o66) instance as the second.
* An object with a postcss method. PostCSS will use that method as described in #2.
* Another [Processor](#38czs75) instance. PostCSS will copy plugins from that instance into this one.

Plugins can also be added by passing them as arguments when creating a postcss instance (see [postcss(plugins)]).

Asynchronous plugins should return a Promise instance.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| plugin | ([Plugin](#488uthg) | [pluginFunction](#2ne53p9) | [Processor](#38czs75)) | PostCSS plugin or [Processor](#38czs75) with plugins. |

#### Returns

#### Processes : Current processor to make methods chain.

Examples

const processor = postcss()  
 .use(autoprefixer)  
 .use(precss)

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L97-L110)

### [**process (css, opts = {})**](#11si5id)

Parses source CSS and returns a [LazyResult](#1ljsd9k) Promise proxy. Because some plugins can be asynchronous it doesn’t make any transformations. Transformations will be applied in the [LazyResult](#1ljsd9k) methods.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| css | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [toString](#12jfdx2) | [Result](#3ls5o66)) | String with input CSS or any object with a toString() method, like a Buffer. Optionally, send a [Result](#3ls5o66) instance and the processor will take the [Root](#2rrrqc1) from it. |
| opts | [processOptions](#29yz7q8)? = {} | Options. |

#### Returns

#### [LazyResult](#1ljsd9k) : Promise proxy.

Examples

processor.process(css, { from: 'a.css', to: 'a.out.css' })  
 .then(result => {  
 console.log(result.css)  
 })

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L17-L170)

### [**Result (processor, root, opts)**](#3ls5o66)

Provides the result of the PostCSS transformations.

A Result instance is returned by [LazyResult#then](#2ce457m) or [Root#toResult](#3w19e94) methods.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| processor | [Processor](#38czs75) | Processor used for this transformation. |
| root | [Root](#2rrrqc1) | Root node after all transformations. |
| opts | [processOptions](#29yz7q8) | Options from the [Processor#process](#11si5id) or [Root#toResult](#3w19e94) . |

#### Examples

postcss([autoprefixer]).process(css).then(result => {  
 console.log(result.css)  
})

const result2 = postcss.parse(css).toResult()

#### Instance Members

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L37-L37)

### [**processor ()**](#20xfydz)

The Processor instance used for this transformation.

Type: [Processor](#38czs75)

#### Examples

for (const plugin of result.processor.plugins) {  
 if (plugin.postcssPlugin === 'postcss-bad') {  
 throw 'postcss-good is incompatible with postcss-bad'  
 }  
})

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L56-L56)

### [**messages ()**](#4kx3h1s)

Contains messages from plugins (e.g., warnings or custom messages). Each message should have type and plugin properties.

Type: [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Message](#p49hy1)>

#### Examples

postcss.plugin('postcss-min-browser', () => {  
 return (root, result) => {  
 const browsers = detectMinBrowsersByCanIUse(root)  
 result.messages.push({  
 type: 'min-browser',  
 plugin: 'postcss-min-browser',  
 browsers  
 })  
 }  
})

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L65-L65)

### [**root ()**](#302dr9l)

Root node after all transformations.

Type: [Root](#2rrrqc1)

#### Examples

root.toResult().root === root

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L75-L75)

### [**opts ()**](#1f7o1he)

Options from the [Processor#process](#11si5id) or [Root#toResult](#3w19e94) call that produced this Result instance.

Type: [processOptions](#29yz7q8)

#### Examples

root.toResult(opts).opts === opts

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L84-L84)

### [**css ()**](#3z7bk57)

A CSS string representing of [Result#root](#302dr9l).

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

postcss.parse('a{}').toResult().css //=> "a{}"

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L99-L99)

### [**map ()**](#2eclud0)

An instance of SourceMapGenerator class from the source-map library, representing changes to the [Result#root](#302dr9l) instance.

Type: SourceMapGenerator

#### Examples

result.map.toJSON() //=> { version: 3, file: 'a.css', … }

if (result.map) {  
 fs.writeFileSync(result.opts.to + '.map', result.map.toString())  
}

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L110-L112)

### [**toString ()**](#thw4kt)

Returns for @{link Result#css} content.

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : String representing of [Result#root](#302dr9l) .

Examples

result + '' === result.css

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L130-L141)

### [**warn (text, opts = {})**](#3dhjn8m)

Creates an instance of [Warning](#46ad4c2) and adds it to [Result#messages](#4kx3h1s).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| text | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Warning message. |
| opts | [Object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Warning options. |
| opts.node | [Node](#1qoc8b1) | CSS node that caused the warning. |
| opts.word | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Word in CSS source that caused the warning. |
| opts.index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | Index in CSS node string that caused the warning. |
| opts.plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Name of the plugin that created this warning. [Result#warn](#3dhjn8m) fills this property automatically. |

#### Returns

[Warning](#46ad4c2) : Created warning.

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L154-L156)

### [**warnings ()**](#1smtxgf)

Returns warnings from plugins. Filters [Warning](#46ad4c2) instances from [Result#messages](#4kx3h1s).

#### Returns

#### [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Warning](#46ad4c2)> : Warnings from plugins.

Examples

result.warnings().forEach(warn => {  
 console.warn(warn.toString())  
})

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L167-L169)

### [**content ()**](#4cmhg48)

An alias for the [Result#css](#3z7bk57) property. Use it with syntaxes that generate non-CSS output.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

result.css === result.content

Extends [Container](#1hmsyys)  [lib/root.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/root.js#L13-L116)

### [**Root (defaults)**](#2rrrqc1)

Represents a CSS file and contains all its parsed nodes.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | any |  |

#### Examples

const root = postcss.parse('a{color:black} b{z-index:2}')  
root.type //=> 'root'  
root.nodes.length //=> 2

#### Instance Members

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L322-L331)

### [**append (children)**](#16x20ju)

Inserts new nodes to the end of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.append(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L63-L89)

### [**each (callback)**](#3qwpj7n)

Iterates through the container’s immediate children, calling callback for each child.

Returning false in the callback will break iteration.

This method only iterates through the container’s immediate children. If you need to recursively iterate through all the container’s descendant nodes, use [Container#walk](#2r0uhxc).

Unlike the for {}-cycle or Array#forEach this iterator is safe if you are mutating the array of child nodes during iteration. PostCSS will adjust the current index to match the mutations.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

const root = postcss.parse('a { color: black; z-index: 1 }')  
const rule = root.first  
  
for (const decl of rule.nodes) {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Cycle will be infinite, because cloneBefore moves the current node  
 // to the next index  
}  
  
rule.each(decl => {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Will be executed only for color and z-index  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L539-L541)

### [**every (condition)**](#261ztfg)

Returns true if callback returns true for all of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is every child pass condition.

Examples

const noPrefixes = rule.every(i => i.prop[0] !== '-')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L582-L585)

### [**first ()**](#l7a3n9)

The container’s first child.

Type: [Node](#1qoc8b1)

#### Examples

rule.first === rules.nodes[0]

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L568-L572)

### [**index (child)**](#356xmb2)

Returns a child’s index within the [Container#nodes](http://docs.google.com/Container#nodes) array.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | [Node](#1qoc8b1) | Child of the current container. |

#### Returns

#### [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) : Child index.

Examples

rule.index( rule.nodes[2] ) //=> 2

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L414-L431)

### [**insertAfter (exist, add)**](#1kc7wiv)

Insert new node after old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

[Node](#1qoc8b1) : This node for methods chain.

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L386-L404)

### [**insertBefore (exist, add)**](#44bvf6o)

Insert new node before old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.insertBefore(decl, decl.clone({ prop: '-webkit-' + decl.prop }))

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L595-L598)

### [**last ()**](#2jh5peh)

The container’s last child.

Type: [Node](#1qoc8b1)

#### Examples

rule.last === rule.nodes[rule.nodes.length - 1]

[lib/root.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/root.js#L95-L99)

### [**on (event, visitor?, type?)**](#ymfzma)

Add visitor for next PostCSS walk.

Visitor subscribes for events. Each event contain node type (atrule, rule, decl, comment) and phase (enter, exit) separated with dot. The default phase is enter. As result possible events could be like comment.enter, decl.exit or rule (equal to rule.enter).

PostCSS will walk through CSS AST and call visitor according current node. Visitor will receive node and node’s index.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| event | any |  |
| visitor | [visitor](#3mj2wkv)? | Function receives node and index. |
| type | visitingEvent? | The type of the node and phase. |

#### Returns

#### [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined) :

Examples

css.on('decl', (node, index) => {  
 if (node.prop === 'will-change') {  
 node.cloneBefore({ prop: 'backface-visibility', value: 'hidden' })  
 }  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L353-L366)

### [**prepend (children)**](#3im3ia3)

Inserts new nodes to the start of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.prepend(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L475-L482)

### [**removeAll ()**](#1xrdshw)

Removes all children from the container and cleans their parent properties.

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.removeAll()  
rule.nodes.length //=> 0

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L447-L463)

### [**removeChild (child)**](#4hr1b5p)

Removes node from the container and cleans the parent properties from the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain

Examples

rule.nodes.length //=> 5  
rule.removeChild(decl)  
rule.nodes.length //=> 4  
decl.parent //=> undefined

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L510-L526)

### [**replaceValues (pattern, opts, callback)**](#2wwbldi)

Passes all declaration values within the container that match pattern through callback, replacing those values with the returned result of callback.

This method is useful if you are using a custom unit or function and need to iterate through all values.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| pattern | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp)) | Replace pattern. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | Options to speed up the search. |
| opts.props | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)>) | An array of property names. |
| opts.fast | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | String that’s used to narrow down values and speed up the regexp search. |
| callback | ([function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)) | String to replace pattern or callback that returns a new value. The callback will receive the same arguments as those passed to a function parameter of String#replace . |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

root.replaceValues(/\d+rem/, { fast: 'rem' }, string => {  
 return 15 \* parseInt(string) + 'px'  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L554-L556)

### [**some (condition)**](#1c1lvlb)

Returns true if callback returns true for (at least) one of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is some child pass condition.

Examples

const hasPrefix = rule.some(i => i.prop[0] === '-')

[lib/root.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/root.js#L64-L70)

### [**toResult (opts = {})**](#3w19e94)

Returns a [Result](#3ls5o66) instance representing the root’s CSS.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| opts | [processOptions](#29yz7q8)? = {} | Options with only to and map keys. |

#### Returns

#### [Result](#3ls5o66) : Result with current root’s CSS.

Examples

const root1 = postcss.parse(css1, { from: 'a.css' })  
const root2 = postcss.parse(css2, { from: 'b.css' })  
root1.append(root2)  
const result = root1.toResult({ to: 'all.css', map: true })

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L110-L124)

### [**walk (callback)**](#2b6jogx)

Traverses the container’s descendant nodes, calling callback for each node.

Like container.each(), this method is safe to use if you are mutating arrays during iteration.

If you only need to iterate through the container’s immediate children, use [Container#each](#19c6y18).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walk(node => {  
 // Traverses all descendant nodes.  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L255-L276)

### [**walkAtRules (name?, callback)**](#qbtyoq)

Traverses the container’s descendant nodes, calling callback for each at-rule node.

If you pass a filter, iteration will only happen over at-rules that have matching names.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| name | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter at-rules by name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkAtRules(rule => {  
 if (isOld(rule.name)) rule.remove()  
})  
  
let first = false  
root.walkAtRules('charset', rule => {  
 if (!first) {  
 first = true  
 } else {  
 rule.remove()  
 }  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L294-L300)

### [**walkComments (callback)**](#3abhhcj)

Traverses the container’s descendant nodes, calling callback for each comment node.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkComments(comment => {  
 comment.remove()  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L155-L176)

### [**walkDecls (prop?, callback)**](#1pgrrkc)

Traverses the container’s descendant nodes, calling callback for each declaration node.

If you pass a filter, iteration will only happen over declarations with matching properties.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter declarations by property name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkDecls(decl => {  
 checkPropertySupport(decl.prop)  
})  
  
root.walkDecls('border-radius', decl => {  
 decl.remove()  
})  
  
root.walkDecls(/^background/, decl => {  
 decl.value = takeFirstColorFromGradient(decl.value)  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L201-L223)

### [**walkRules (selector?, callback)**](#49gfa85)

Traverses the container’s descendant nodes, calling callback for each rule node.

If you pass a filter, iteration will only happen over rules with matching selectors.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| selector | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter rules by selector. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : returns false if iteration was broke.

Examples

const selectors = []  
root.walkRules(rule => {  
 selectors.push(rule.selector)  
})  
console.log(`Your CSS uses ${ selectors.length } selectors`)

Extends [Container](#1hmsyys)  [lib/rule.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/rule.js#L15-L87)

### [**Rule (defaults)**](#2olpkfy)

Represents a CSS rule: a selector followed by a declaration block.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | any |  |

#### Examples

const root = postcss.parse('a{}')  
const rule = root.first  
rule.type //=> 'rule'  
rule.toString() //=> 'a{}'

#### Instance Members

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L322-L331)

### [**append (children)**](#13qzunr)

Inserts new nodes to the end of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.append(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L63-L89)

### [**each (callback)**](#3nqndbk)

Iterates through the container’s immediate children, calling callback for each child.

Returning false in the callback will break iteration.

This method only iterates through the container’s immediate children. If you need to recursively iterate through all the container’s descendant nodes, use [Container#walk](#2r0uhxc).

Unlike the for {}-cycle or Array#forEach this iterator is safe if you are mutating the array of child nodes during iteration. PostCSS will adjust the current index to match the mutations.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

const root = postcss.parse('a { color: black; z-index: 1 }')  
const rule = root.first  
  
for (const decl of rule.nodes) {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Cycle will be infinite, because cloneBefore moves the current node  
 // to the next index  
}  
  
rule.each(decl => {  
 decl.cloneBefore({ prop: '-webkit-' + decl.prop })  
 // Will be executed only for color and z-index  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L539-L541)

### [**every (condition)**](#22vxnjd)

Returns true if callback returns true for all of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is every child pass condition.

Examples

const noPrefixes = rule.every(i => i.prop[0] !== '-')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L582-L585)

### [**first ()**](#i17xr6)

The container’s first child.

Type: [Node](#1qoc8b1)

#### Examples

rule.first === rules.nodes[0]

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L568-L572)

### [**index (child)**](#320vgez)

Returns a child’s index within the [Container#nodes](http://docs.google.com/Container#nodes) array.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | [Node](#1qoc8b1) | Child of the current container. |

#### Returns

#### [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) : Child index.

Examples

rule.index( rule.nodes[2] ) //=> 2

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L414-L431)

### [**insertAfter (exist, add)**](#1h65qms)

Insert new node after old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

[Node](#1qoc8b1) : This node for methods chain.

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L386-L404)

### [**insertBefore (exist, add)**](#415t9al)

Insert new node before old node within the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| exist | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |
| add | ([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New node. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.insertBefore(decl, decl.clone({ prop: '-webkit-' + decl.prop }))

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L595-L598)

### [**last ()**](#2gb3jie)

The container’s last child.

Type: [Node](#1qoc8b1)

#### Examples

rule.last === rule.nodes[rule.nodes.length - 1]

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L353-L366)

### [**prepend (children)**](#vgdtq7)

Inserts new nodes to the start of the container.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| children | ...([Node](#1qoc8b1) | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)>) | New nodes. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

const decl1 = postcss.decl({ prop: 'color', value: 'black' })  
const decl2 = postcss.decl({ prop: 'background-color', value: 'white' })  
rule.prepend(decl1, decl2)  
  
root.append({ name: 'charset', params: '"UTF-8"' }) // at-rule  
root.append({ selector: 'a' }) // rule  
rule.append({ prop: 'color', value: 'black' }) // declaration  
rule.append({ text: 'Comment' }) // comment  
  
root.append('a {}')  
root.first.append('color: black; z-index: 1')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L475-L482)

### [**removeAll ()**](#3fg1ce0)

Removes all children from the container and cleans their parent properties.

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

rule.removeAll()  
rule.nodes.length //=> 0

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L447-L463)

### [**removeChild (child)**](#1ulbmlt)

Removes node from the container and cleans the parent properties from the node and its children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| child | ([Node](#1qoc8b1) | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)) | Child or child’s index. |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain

Examples

rule.nodes.length //=> 5  
rule.removeChild(decl)  
rule.nodes.length //=> 4  
decl.parent //=> undefined

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L510-L526)

### [**replaceValues (pattern, opts, callback)**](#4ekz59m)

Passes all declaration values within the container that match pattern through callback, replacing those values with the returned result of callback.

This method is useful if you are using a custom unit or function and need to iterate through all values.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| pattern | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp)) | Replace pattern. |
| opts | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | Options to speed up the search. |
| opts.props | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)>) | An array of property names. |
| opts.fast | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | String that’s used to narrow down values and speed up the regexp search. |
| callback | ([function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function) | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)) | String to replace pattern or callback that returns a new value. The callback will receive the same arguments as those passed to a function parameter of String#replace . |

#### Returns

#### [Node](#1qoc8b1) : This node for methods chain.

Examples

root.replaceValues(/\d+rem/, { fast: 'rem' }, string => {  
 return 15 \* parseInt(string) + 'px'  
})

[lib/rule.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/rule.js#L38-L40)

### [**selectors ()**](#2tq9fhf)

An array containing the rule’s individual selectors. Groups of selectors are split at commas.

Type: [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)>

#### Examples

const root = postcss.parse('a, b { }')  
const rule = root.first  
  
rule.selector //=> 'a, b'  
rule.selectors //=> ['a', 'b']  
  
rule.selectors = ['a', 'strong']  
rule.selector //=> 'a, strong'

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L554-L556)

### [**some (condition)**](#18vjpp8)

Returns true if callback returns true for (at least) one of the container’s children.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| condition | [childCondition](#4gjguf0) | Iterator returns true or false. |

#### Returns

#### [boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) : Is some child pass condition.

Examples

const hasPrefix = rule.some(i => i.prop[0] === '-')

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L110-L124)

### [**walk (callback)**](#3sv78d1)

Traverses the container’s descendant nodes, calling callback for each node.

Like container.each(), this method is safe to use if you are mutating arrays during iteration.

If you only need to iterate through the container’s immediate children, use [Container#each](#19c6y18).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walk(node => {  
 // Traverses all descendant nodes.  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L255-L276)

### [**walkAtRules (name?, callback)**](#280hiku)

Traverses the container’s descendant nodes, calling callback for each at-rule node.

If you pass a filter, iteration will only happen over at-rules that have matching names.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| name | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter at-rules by name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkAtRules(rule => {  
 if (isOld(rule.name)) rule.remove()  
})  
  
let first = false  
root.walkAtRules('charset', rule => {  
 if (!first) {  
 first = true  
 } else {  
 rule.remove()  
 }  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L294-L300)

### [**walkComments (callback)**](#n5rssn)

Traverses the container’s descendant nodes, calling callback for each comment node.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkComments(comment => {  
 comment.remove()  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L155-L176)

### [**walkDecls (prop?, callback)**](#375fbgg)

Traverses the container’s descendant nodes, calling callback for each declaration node.

If you pass a filter, iteration will only happen over declarations with matching properties.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter declarations by property name. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returns false if iteration was broke.

Examples

root.walkDecls(decl => {  
 checkPropertySupport(decl.prop)  
})  
  
root.walkDecls('border-radius', decl => {  
 decl.remove()  
})  
  
root.walkDecls(/^background/, decl => {  
 decl.value = takeFirstColorFromGradient(decl.value)  
})

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L201-L223)

### [**walkRules (selector?, callback)**](#1maplo9)

Traverses the container’s descendant nodes, calling callback for each rule node.

If you pass a filter, iteration will only happen over rules with matching selectors.

Like [Container#each](#19c6y18), this method is safe to use if you are mutating arrays during iteration.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| selector | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [RegExp](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/RegExp))? | String or regular expression to filter rules by selector. |
| callback | [childIterator](#1wjtbr7) | Iterator receives each node and index. |

#### Returns

#### (false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : returns false if iteration was broke.

Examples

const selectors = []  
root.walkRules(rule => {  
 selectors.push(rule.selector)  
})  
console.log(`Your CSS uses ${ selectors.length } selectors`)

[lib/warning.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/warning.js#L9-L107)

### [**Warning (text, opts = {})**](#46ad4c2)

Represents a plugin’s warning. It can be created using [Node#warn](#2pta16n).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| text | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Warning message. |
| opts | [Object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? = {} | Warning options. |

#### Examples

if (decl.important) {  
 decl.warn(result, 'Avoid !important', { word: '!important' })  
}

#### Instance Members

[lib/warning.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/warning.js#L31-L31)

### [**type ()**](#2lfnejv)

Type to filter warnings from [Result#messages](#4kx3h1s). Always equal to "warning".

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

const nonWarning = result.messages.filter(i => i.type !== 'warning')

[lib/warning.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/warning.js#L40-L40)

### [**text ()**](#10kxoro)

The warning message.

Type: [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)

#### Examples

warning.text //=> 'Try to avoid !important'

[lib/warning.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/warning.js#L51-L51)

### [**line ()**](#3kkl7fh)

Line in the input file with this warning’s source.

Type: [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)

#### Examples

warning.line //=> 5

[lib/warning.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/warning.js#L60-L60)

### [**column ()**](#1zpvhna)

Column in the input file with this warning’s source.

Type: [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number)

#### Examples

warning.column //=> 6

[lib/warning.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/warning.js#L74-L88)

### [**toString ()**](#4jpj0b3)

Returns a warning position and message.

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : Warning position and message.

Examples

warning.toString() //=> 'postcss-lint:a.css:10:14: Avoid !important'

NAMESPACES

[lib/list.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/list.js#L10-L85)

### [**list ()**](#1e03kqp)

Contains helpers for safely splitting lists of CSS values, preserving parentheses and quotes.

#### Examples

const list = postcss.list

#### Static Members

[lib/list.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/list.js#L64-L67)

### [**space (string)**](#3xzr3ei)

Safely splits space-separated values (such as those for background, border-radius, and other shorthand properties).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| string | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Space-separated values. |

#### Returns

#### [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)> : Split values.

Examples

postcss.list.space('1px calc(10% + 1px)') //=> ['1px', 'calc(10% + 1px)']

[lib/list.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/list.js#L81-L83)

### [**comma (string)**](#2d51dmb)

Safely splits comma-separated values (such as those for transition-\* and background properties).

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| string | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Comma-separated values. |

#### Returns

#### [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String)> : Split values.

Examples

postcss.list.comma('black, linear-gradient(white, black)')  
//=> ['black', 'linear-gradient(white, black)']

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L30-L35)

### [**postcss (plugins)**](#sabnu4)

Create a new [Processor](#38czs75) instance that will apply plugins as CSS processors.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| plugins | ([Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<([Plugin](#488uthg) | [pluginFunction](#2ne53p9))> | [Processor](#38czs75)) | PostCSS plugins. See [Processor#use](#2mn7vak) for plugin format. |

#### Returns

#### [Processor](#38czs75) : Processor to process multiple CSS.

Examples

let postcss = require('postcss')  
  
postcss(plugins).process(css, { from, to }).then(result => {  
 console.log(result.css)  
})

#### Static Members

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L109-L130)

### [**plugin (name, initializer)**](#3c9z6hx)

Creates a PostCSS plugin with a standard API.

The newly-wrapped function will provide both the name and PostCSS version of the plugin.

const processor = postcss([replace])  
processor.plugins[0].postcssPlugin //=> 'postcss-replace'  
processor.plugins[0].postcssVersion //=> '6.0.0'

The plugin function receives 2 arguments: [Root](#2rrrqc1) and [Result](#3ls5o66) instance. The function should mutate the provided Root node. Alternatively, you can create a new Root node and override the result.root property.

const cleaner = postcss.plugin('postcss-cleaner', () => {  
 return (root, result) => {  
 result.root = postcss.root()  
 }  
})

As a convenience, plugins also expose a process method so that you can use them as standalone tools.

cleaner.process(css, processOpts, pluginOpts)  
// This is equivalent to:  
postcss([ cleaner(pluginOpts) ]).process(css, processOpts)

Asynchronous plugins should return a Promise instance.

postcss.plugin('postcss-import', () => {  
 return (root, result) => {  
 return new Promise( (resolve, reject) => {  
 fs.readFile('base.css', (base) => {  
 root.prepend(base)  
 resolve()  
 })  
 })  
 }  
})

Add warnings using the [Node#warn](#2pta16n) method. Send data to other plugins using the [Result#messages](#4kx3h1s) array.

postcss.plugin('postcss-caniuse-test', () => {  
 return (root, result) => {  
 root.walkDecls(decl => {  
 if (!caniuse.support(decl.prop)) {  
 decl.warn(result, 'Some browsers do not support ' + decl.prop)  
 }  
 })  
 }  
})

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| name | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | PostCSS plugin name. Same as in name property in package.json . It will be saved in plugin.postcssPlugin property. |
| initializer | [function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function) | Will receive plugin options and should return [pluginFunction](#2ne53p9) |

#### Returns

[Plugin](#488uthg) : PostCSS plugin.

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L143-L143)

### [**stringify (node, builder)**](#1rf9gpq)

Default function to convert a node tree into a CSS string.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| node | [Node](#1qoc8b1) | Start node for stringifing. Usually [Root](#2rrrqc1) . |
| builder | [builder](#21od6so) | Function to concatenate CSS from node’s parts or generate string and source map. |

#### Returns

void :

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L163-L163)

### [**parse (css, opts?)**](#4bewzdj)

Parses source css and returns a new [Root](#2rrrqc1) node, which contains the source CSS nodes.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| css | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [toString](#12jfdx2)) | String with input CSS or any object with toString() method, like a Buffer |
| opts | [processOptions](#29yz7q8)? | Options with only from and map keys. |

#### Returns

#### [Root](#2rrrqc1) : PostCSS AST.

Examples

// Simple CSS concatenation with source map support  
const root1 = postcss.parse(css1, { from: file1 })  
const root2 = postcss.parse(css2, { from: file2 })  
root1.append(root2).toResult().css

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L173-L173)

### [**vendor ()**](#2qk79lc)

Contains the [vendor](#434ayfz) module.

Type: [vendor](#434ayfz)

#### Examples

postcss.vendor.unprefixed('-moz-tab') //=> ['tab']

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L183-L183)

### [**list ()**](#15phjt5)

Contains the [list](#1e03kqp) module.

Type: [list](#1e03kqp)

#### Examples

postcss.list.space('5px calc(10% + 5px)') //=> ['5px', 'calc(10% + 5px)']

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L195-L195)

### [**comment (defaults?)**](#3pp52gy)

Creates a new [Comment](#1y810tw) node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Properties for the new node. |

#### Returns

#### [Comment](#1y810tw) : New comment node

Examples

postcss.comment({ text: 'test' })

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L207-L207)

### [**atRule (defaults?)**](#24ufcor)

Creates a new [AtRule](#30j0zll) node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Properties for the new node. |

#### Returns

#### [AtRule](#30j0zll) : new at-rule node

Examples

postcss.atRule({ name: 'charset' }).toString() //=> "@charset"

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L219-L219)

### [**decl (defaults?)**](#jzpmwk)

Creates a new [Declaration](#pkwqa1) node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Properties for the new node. |

#### Returns

#### [Declaration](#pkwqa1) : new declaration node

Examples

postcss.decl({ prop: 'color', value: 'red' }).toString() //=> "color: red"

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L231-L231)

### [**rule (defaults?)**](#33zd5kd)

Creates a new [Rule](#2olpkfy) node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Properties for the new node. |

#### Returns

#### [Rule](#2olpkfy) : new rule node

Examples

postcss.rule({ selector: 'a' }).toString() //=> "a {\n}"

[lib/postcss.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/postcss.js#L243-L243)

### [**root (defaults?)**](#1j4nfs6)

Creates a new [Root](#2rrrqc1) node.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| defaults | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)? | Properties for the new node. |

#### Returns

#### [Root](#2rrrqc1) : new root node.

Examples

postcss.root({ after: '\n' }).toString() //=> "\n"

[lib/vendor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/vendor.js#L9-L45)

### [**vendor ()**](#434ayfz)

Contains helpers for working with vendor prefixes.

#### Examples

const vendor = postcss.vendor

#### Static Members

[lib/vendor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/vendor.js#L22-L29)

### [**prefix (prop)**](#2i9l8ns)

Returns the vendor prefix extracted from an input string.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | String with or without vendor prefix. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : vendor prefix or empty string

Examples

postcss.vendor.prefix('-moz-tab-size') //=> '-moz-'  
postcss.vendor.prefix('tab-size') //=> ''

[lib/vendor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/vendor.js#L41-L43)

### [**unprefixed (prop)**](#xevivl)

Returns the input string stripped of its vendor prefix.

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| prop | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | String with or without vendor prefix. |

#### Returns

#### [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) : String name without vendor prefixes.

Examples

postcss.vendor.unprefixed('-moz-tab-size') //=> 'tab-size'

GLOBAL

[lib/result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/result.js#L174-L178)

### [**Message ()**](#p49hy1)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| type | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | : Message type. |
| plugin | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | : Source PostCSS plugin name. |

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L183-L186)

### [**Plugin ()**](#488uthg)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| postcss | [function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function) | : PostCSS plugin function. |

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L139-L144)

### [**builder (part, node, type?)**](#21od6so)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| part | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | Part of generated CSS connected to this node. |
| node | [Node](#1qoc8b1) | AST node. |
| type | ("start" | "end")? | Node’s part type. |

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L697-L703)

### [**childCondition (node, index, nodes)**](#4gjguf0)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| node | [Node](#1qoc8b1) | Container child. |
| index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | Child index. |
| nodes | [Array](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Array)<[Node](#1qoc8b1)> | All container children. |

#### Returns

[boolean](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Boolean) :

[lib/container.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/container.js#L705-L710)

### [**childIterator (node, index)**](#1wjtbr7)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| node | [Node](#1qoc8b1) | Container child. |
| index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | Child index. |

#### Returns

(false | [undefined](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/undefined)) : Returning false will break iteration.

[lib/input.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/input.js#L171-L176)

### [**filePosition ()**](#3utoxif)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| file | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | : Path to file. |
| line | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | : Source line in file. |
| column | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | : Source column in file. |

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L431-L434)

### [**onFulfilled (result)**](#393x0lu)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| result | [Result](#3ls5o66) |  |

[lib/lazy-result.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/lazy-result.js#L436-L439)

### [**onRejected (error)**](#1o97atn)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| error | [Error](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Error) |  |

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L146-L154)

### [**parser (css, opts?)**](#gtnh0h)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| css | ([string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | [toString](#12jfdx2)) | String with input CSS or any object with toString() method, like a Buffer. |
| opts | [processOptions](#29yz7q8)? | Options with only from and map keys. |

#### Returns

[Root](#2rrrqc1) : PostCSS AST

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L177-L181)

### [**pluginFunction (root, result)**](#2ne53p9)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| root | [Root](#2rrrqc1) | Parsed input CSS. |
| result | [Result](#3ls5o66) | Result to set warnings or check other plugins. |

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L568-L572)

### [**position ()**](#30tazoa)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| line | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | : Source line in file. |
| column | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | : Source column in file. |

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L188-L219)

### [**processOptions ()**](#29yz7q8)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| from | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | : The path of the CSS source file. You should always set from , because it is used in source map generation and syntax error messages. |
| to | [string](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/String) | : The path where you’ll put the output CSS file. You should always set to to generate correct source maps. |
| parser | [parser](#gtnh0h) | : Function to generate AST by string. |
| stringifier | [stringifier](#2vor4mt) | : Class to generate string by AST. |
| syntax | [syntax](#1au1eum) | : Object with parse and stringify . |
| map | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | : Source map options. |
| map | [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object) | : Source map options. |

[lib/node.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/node.js#L574-L579)

### [**source ()**](#1fyl9w3)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| input | [Input](#4du1wux) | : [Input](#4du1wux) with input file |
| start | [position](#30tazoa) | : The starting position of the node’s source. |
| end | [position](#30tazoa) | : The ending position of the node’s source. |

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L156-L164)

### [**stringifier (node, builder)**](#2vor4mt)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| node | [Node](#1qoc8b1) | Start node for stringifing. Usually [Root](#2rrrqc1) . |
| builder | [builder](#21od6so) | Function to concatenate CSS from node’s parts or generate string and source map. |

#### Returns

void :

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L166-L170)

### [**syntax ()**](#1au1eum)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| parse | [parser](#gtnh0h) | : Function to generate AST by string. |
| stringify | [stringifier](#2vor4mt) | : Function to generate string by AST. |

[lib/processor.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/processor.js#L172-L175)

### [**toString ()**](#12jfdx2)

Type: [object](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object)

|  |  |  |
| --- | --- | --- |
| property | type | description |
| toString | [function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function) |  |

[lib/root.js](https://git@github.com/:postcss/postcss/blob/ae7cf844d806d0e9f536cfd5b5a680070ebed3ce/lib/root.js#L118-L122)

### [**visitor (node, index)**](#3mj2wkv)

Type: [Function](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/function)

|  |  |  |
| --- | --- | --- |
| parameter | type | description |
| node | [Node](#1qoc8b1) | Container child. |
| index | [number](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Number) | Child index. |